https://wiki.micylou.com/index.php/Mediawiki_Extension

Mediawiki Extension

Mediawiki Extension

Source:	Query
Language:	English
Topic:	Mediawiki
SubTopic:	Query
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Table
Status:	Active
Access:	free

Download this page as PDF

Our Knowledge Management Solution based on MEDIAWIKI

This is the result of a semantic query looking for all source documents with the following properties

- Topic::Mediawiki

You have to choose if you want to see the result in a table, listing all documents with their shortcut and related information, or if you wish to create one single document containing all the ones retrieved by the query with an automatic table of content created. This second option is practival when, for example, a technician needs to retrieve all the documentation of a product.

It makes no difference 1, 10, 100 persons maintain those documents or create new ones, this list is automatically updated as it is created using a Semantic Query.

Back to Welcome Page

To view related documents, select an output format in the drop-down list here below.

Result in a limited list (10 records) ▼

Section	Topic	SubTopic	Source	Language	DocumentType	Status
A Knowledge Management Solution based on MEDIAWIKI	Mediawiki	Query	DataSource	English	Documentation	Active
Install/upgrade/Downgrade Composer	Mediawiki	Linux	DataSource	English	Documentation	Active

Mediawiki - Administrator handbook	Mediawiki	Book	DataSource	English	User Guide	Active
Mediawiki - Editor handbook	Mediawiki	Book	DataSource	English	User Guide	Active
Mediawiki - LDAP Stack	Mediawiki	Integration	DataSource	English	User Guide	Active
Mediawiki Additional Namespaces	Mediawiki	Configuration	DataSource	English	Documentation	Active
Mediawiki Extension AdminLinks	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension ClipUpload	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension DrawIO Editor	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension DynamicPageList	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active

... further results

Section	Topic	SubTopic	Source	Language	DocumentType	Status
A Knowledge Management Solution based on MEDIAWIKI	Mediawiki	Query	DataSource	English	Documentation	Active
Install/upgrade/Downgrade Composer	Mediawiki	Linux	DataSource	English	Documentation	Active
Mediawiki - Administrator handbook	Mediawiki	Book	DataSource	English	User Guide	Active
Mediawiki - Editor handbook	Mediawiki	Book	DataSource	English	User Guide	Active
Mediawiki - LDAP Stack	Mediawiki	Integration	DataSource	English	User Guide	Active
Mediawiki Additional Namespaces	Mediawiki	Configuration	DataSource	English	Documentation	Active
Mediawiki Extension AdminLinks	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension ClipUpload	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension DrawIO Editor	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension DynamicPageList	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension EmbedVideo	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension Flex Diagrams	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension Iframe	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension MassEditRegex	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension MsInsert	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active

Mediawiki Extension MsUpload	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension MultiLanguageManager	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension NativeSvgHandler	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension PDF Embed	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension PageForms	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension PageSchema	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension Quiz	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension RenameUser	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension Restrict Access By Category and Group	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension RightFunctions	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension Semantic Mediawiki	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension ShowMe	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension UserMerge	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Extension WhosOnline	Mediawiki	Mediawiki Extension	DataSource	English	Documentation	Active
Mediawiki Security Matrix	Mediawiki	Security	DataSource	English	Documentation	Active
Mediawiki Toolbox Personnalisation	Mediawiki	Configuration	DataSource	English	Documentation	Active
Mediawiki backup script	Mediawiki	Linux	DataSource	English	Documentation	Active
Syntaxe Mediawiki - How-to ?	Mediawiki	Syntax	DataSource	Français	User Guide	Active
WIKI - KMS - Crontabs	Mediawiki	Linux	DataSource	English	Documentation	Active
Working with Mediawiki	Mediawiki	Book	DataSource	English	User Guide	Active

□

Contents

- 1 A Knowledge Management Solution based on MEDIAWIKI
  - 1.1 Presentation
- 2 Install/upgrade/Downgrade Composer
  - 2.1 Installation of Composer
  - 2.2 Upgrade
  - 2.3 Downgrade
  - 2.4 pre-release version

- 3 Mediawiki - Administrator handbook
- 4 Mediawiki - Editor handbook
- 5 Mediawiki - LDAP Stack
- 6 Mediawiki Additional Namespaces
- 7 Mediawiki Extension AdminLinks
  - 7.1 Description
  - 7.2 Installation
    - 7.2.1 Download
    - 7.2.2 Activation
    - 7.2.3 Configuration
    - 7.2.4 Extensions using Admin Links
- 8 Mediawiki Extension ClipUpload
  - 8.1 Description
  - 8.2 Installation
    - 8.2.1 Download
    - 8.2.2 Unzip
    - 8.2.3 Rename folder
    - 8.2.4 Activate the extension
    - 8.2.5 Configuration
- 9 Mediawiki Extension DrawIO Editor
  - 9.1 Description
  - 9.2 Installation
    - 9.2.1 Download
    - 9.2.2 Activation
    - 9.2.3 Usage
    - 9.2.4 Warning
    - 9.2.5 Features
    - 9.2.6 Requirements
    - 9.2.7 Example
- 10 Mediawiki Extension DynamicPageList
  - 10.1 Description
  - 10.2 Installation
    - 10.2.1 Download
    - 10.2.2 Activation
    - 10.2.3 Configuration
    - 10.2.4 Use
    - 10.2.5 Page selection
      - 10.2.5.1 category
      - 10.2.5.2 notcategory
      - 10.2.5.3 namespace
      - 10.2.5.4 redirects
      - 10.2.5.5 stablepages
      - 10.2.5.6 count
    - 10.2.6 Details displayed
      - 10.2.6.1 shownamespace
      - 10.2.6.2 mode=gallery
      - 10.2.6.3 suppresserrors
      - 10.2.6.4 nofollow
      - 10.2.6.5 googlehack
      - 10.2.6.6 Order
      - 10.2.6.7 alphabetical example

- 10.2.6.8 order
- 11 Mediawiki Extension EmbedVideo
  - 11.1 Description
  - 11.2 Installation
    - 11.2.1 Download
    - 11.2.2 Activation
    - 11.2.3 Configuration
    - 11.2.4 Developper note
- 12 Mediawiki Extension Flex Diagrams
  - 12.1 Description
  - 12.2 Installation
    - 12.2.1 Download
    - 12.2.2 Activation
    - 12.2.3 Configuration
    - 12.2.4 Usage
      - 12.2.4.1 Creating diagrams
      - 12.2.4.2 Adding links to BPMN diagrams
      - 12.2.4.3 Displaying diagrams
    - 12.2.5 Screenshots
    - 12.2.6 See also
- 13 Mediawiki Extension Iframe
  - 13.1 Copyright & security issues
  - 13.2 Known problems
  - 13.3 Installation
    - 13.3.1 Clone the GIT repository
    - 13.3.2 Adding other web addresses
    - 13.3.3 Category
    - 13.3.4 Width and height of an iframe
    - 13.3.5 Delay of iframe loading
- 14 Mediawiki Extension MassEditRegex
  - 14.1 Description
  - 14.2 Installation
    - 14.2.1 Download
    - 14.2.2 Activation
    - 14.2.3 Configuration
    - 14.2.4 Known issue
    - 14.2.5 Usage
    - 14.2.6 See also
- 15 Mediawiki Extension MsInsert
  - 15.1 Description
  - 15.2 Installation
    - 15.2.1 Download
    - 15.2.2 Activation
    - 15.2.3 Configuration
    - 15.2.4 See also
- 16 Mediawiki Extension MsUpload
  - 16.1 Description
  - 16.2 Installation
    - 16.2.1 Download
    - 16.2.2 Activation
    - 16.2.3 Configuration

- 16.2.3.1 Features
- 16.2.3.2 Relevant global configuration variables
- 16.2.4 Remark
- 16.2.5 See also

- 17 Mediawiki Extension MultiLanguageManager
  - 17.1 Description
  - 17.2 Installation
    - 17.2.1 Download
    - 17.2.2 Activation
    - 17.2.3 Configuration
      - 17.2.3.1 Available languages
      - 17.2.3.2 User rights
- 18 Mediawiki Extension NativeSvgHandler
  - 18.1 Description
  - 18.2 Installation
    - 18.2.1 Download
    - 18.2.2 Activation
    - 18.2.3 Security
- 19 Mediawiki Extension PDF Embed
  - 19.1 Description
  - 19.2 Installation
    - 19.2.1 Download
    - 19.2.2 Activation
    - 19.2.3 Configuration
    - 19.2.4 Usage
    - 19.2.5 See also
- 20 Mediawiki Extension PageForms
  - 20.1 Description
  - 20.2 Installation
    - 20.2.1 Download
    - 20.2.2 Activation
    - 20.2.3 Configuration
  - 20.3 Quick start guide
    - 20.3.1 The easy way - Special:CreateClass
    - 20.3.2 Example
  - 20.4 Example
  - 20.5 Using Page Schemas
- 21 Mediawiki Extension PageSchema
  - 21.1 Description
  - 21.2 Installation
    - 21.2.1 Download
    - 21.2.2 Activation
    - 21.2.3 Configuration
    - 21.2.4 Tutorial video
    - 21.2.5 Sample XML structure
- 22 Mediawiki Extension Quiz
  - 22.1 Description
  - 22.2 Installation
    - 22.2.1 Download
    - 22.2.2 Activation
  - 22.3 Create your Quiz

        - 22.3.1 Adding a new question type
    - 22.4 Quiz documentation
    - 22.5 Overview
        - 22.5.1 Version
        - 22.5.2 Pedagogical considerations
    - 22.6 How to create a quiz
        - 22.6.1 The quiz tag
        - 22.6.2 Inserting questions
        - 22.6.3 How to use the examples
            - 22.6.3.1 Copying and pasting
            - 22.6.3.2 Live demos
    - 22.7 Basic question types
        - 22.7.1 Multiple choice / single response
        - 22.7.2 True/false questions
        - 22.7.3 Multiple choice / multiple response
        - 22.7.4 Typed response (short answer)
        - 22.7.5 Gapfill / cloze
        - 22.7.6 Numerical response
        - 22.7.7 Tabular response type with radio buttons
        - 22.7.8 Tabular response type with checkboxes
    - 22.8 Advanced stuff
        - 22.8.1 Inserting feedback
        - 22.8.2 Inserting blocks of text between questions
        - 22.8.3 Using images as answers
        - 22.8.4 Using images in the question
        - 22.8.5 Using animations
        - 22.8.6 Hyperlinks and other markup
            - 22.8.6.1 Markup that works inside quizzes
            - 22.8.6.2 Markup that partly works inside quizzes
        - 22.8.7 Adding music and sound effects
        - 22.8.8 Configuring points
        - 22.8.9 Shuffle Control Tags
    - 22.9 Troubleshooting
    - 22.10 Future stuff
- 23 Mediawiki Extension RenameUser
    - 23.1 Description
    - 23.2 Installation
        - 23.2.1 Download
        - 23.2.2 Activation
        - 23.2.3 Configuration
        - 23.2.4 Usage
        - 23.2.5 See also
- 24 Mediawiki Extension Restrict Access By Category and Group
    - 24.1 Description
    - 24.2 Usage
    - 24.3 Download instructions
    - 24.4 Installation
        - 24.4.1 Folder creation
        - 24.4.2 Create the extension php file
        - 24.4.3 Edit Localsettings.php
            - 24.4.3.1 Activate the extension

- 24.4.3.2 Add the security groups
- 25 Mediawiki Extension RightFunctions
  - 25.1 Description
  - 25.2 Installation
    - 25.2.1 Download
    - 25.2.2 Activation
    - 25.2.3 Configuration
  - 25.3 Usage
    - 25.3.1 Ifright
    - 25.3.2 Ifallowed
    - 25.3.3 Switchright
    - 25.3.4 Userrights
    - 25.3.5 Ifgroup
    - 25.3.6 Switchgroup
    - 25.3.7 Ifpageright
    - 25.3.8 Ifpageallowed
    - 25.3.9 Ifprotected
    - 25.3.10 Getrestrictions
- 26 Mediawiki Extension Semantic Mediawiki
  - 26.1 Pre-requisite
  - 26.2 Installation
  - 26.3 Enabling Semantic Mediawiki
  - 26.4 Setup and update
  - 26.5 Verify
  - 26.6 Testing
- 27 Mediawiki Extension ShowMe
  - 27.1 Description
  - 27.2 Installation
    - 27.2.1 Download
    - 27.2.2 Activation
    - 27.2.3 Usage
    - 27.2.4 Example
- 28 Mediawiki Extension UserMerge
  - 28.1 Description
  - 28.2 Installation
    - 28.2.1 Download
    - 28.2.2 Activation
    - 28.2.3 Configuration
- 29 Mediawiki Extension WhosOnline
  - 29.1 Description
  - 29.2 Installation
    - 29.2.1 Download
    - 29.2.2 Activation
    - 29.2.3 Configuration
    - 29.2.4 Usage
- 30 Mediawiki Security Matrix
  - 30.1 Micylou Standard Mediawiki Security Matrix
    - 30.1.1 Defaults
    - 30.1.2 Roles by groups
    - 30.1.3 Security matrix of our default solution set up in LocalSettings.php - restricted access

- 31 Mediawiki Toolbox Personnalisation
- 32 Mediawiki backup script
- 33 Syntaxe Mediawiki - How-to ?
  - 33.1 Table des matières
  - 33.2 Mise en page
    - 33.2.1 Chapitres, sections, paragraphes
    - 33.2.2 Retour à la ligne
    - 33.2.3 Les listes
    - 33.2.4 Effets divers
  - 33.3 Mise en forme des caractères
  - 33.4 Liens et adresses internet
    - 33.4.1 Les liens
      - 33.4.1.1 Liens internes
      - 33.4.1.2 Liens externes
    - 33.4.2 Les points d'ancrage
    - 33.4.3 Les redirections
    - 33.4.4 Les adresses internet
  - 33.5 Les images
    - 33.5.1 Syntaxe
  - 33.6 Tableaux
    - 33.6.1 Petits exemples
    - 33.6.2 La syntaxe des tableaux
      - 33.6.2.1 table
      - 33.6.2.2 td
      - 33.6.2.3 tr
      - 33.6.2.4 th
      - 33.6.2.5 caption
      - 33.6.2.6 Exemple récapitulatif
  - 33.7 Tableaux imbriqués
    - 33.7.1 Les paramètres
    - 33.7.2 border =
      - 33.7.2.1 rowspan et colspan
      - 33.7.2.2 align et valign
      - 33.7.2.3 cellspacing et cellpadding
      - 33.7.2.4 width et height
      - 33.7.2.5 bgcolor
      - 33.7.2.6 style
    - 33.7.3 Remarques
      - 33.7.3.1 Quand les tableaux sont-ils appropriés ?
      - 33.7.3.2 Quand les tableaux sont-ils inappropriés ?
        - 33.7.3.2.1 Liste très longues ou listes très courtes
        - 33.7.3.2.2 Légende d'illustration
      - 33.7.3.3 Problèmes éventuels
    - 33.7.4 synthèse / aide-mémoire
  - 33.8 Commentaire
  - 33.9 Catégories
- 34 WIKI - KMS - Crontabs
  - 34.1 RunJobs.sh
    - 34.1.1 Code script RunJobs.sh
  - 34.2 Indexation.sh
    - 34.2.1 Script Indexation.sh

- 34.3 BuildSemanticData.sh
  - 34.3.1 Code script BuildSemanticData.sh
- 34.4 Crontab script
  - 34.4.1 Code script ScriptingBulk.sh
- 34.5 Scheduling the scripts to run
- 34.6 Basic commands
- 34.7 crontab syntax
- 35 Working with Mediawiki

A Knowledge Management Solution based on MEDIAWIKI

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Query
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Presentation

Welcome on our Knowledge Management System platform based on the Mediawiki Long Term Service version (LTS). Mediawiki is the engine running Wikipedia, the well-known free encyclopedia on the web, as well many others.

This wiki not only serves as a demo of our solution but is also our own KMS; we have taken the opportunity to separate confidential and/restricted information from the publicly accessible information. Our solution offers not only document protected information, but we can also protect only parts of documents.

We therefore offer you the real possibility to have only **one source** of information allowing / restricting access to the information exactly as you need it.

This WIKI shows you the ability to automatically create documents based on semantic content, this independently of the number of source documents or the number of the different editors having created them.

You can see one sample from **this Mediawiki page** querying all documents in this wiki using a Semantic Query, offering you a table list or embedding all source documents into a new one, right on your screen.

Our solutions is **entirely based on free opensource products**. We dit put them together based on our experience to propose the best starting point possible to your requirements; we believe we

already cover most of them. This solution at the contrary of most wikis embeded in other solutions like ITSM tools, ERP tools, ... is NOT limited to basic functionalities but goes way beyond and contains its own documentation for each component installed and/or configured.

Back to top of page - Back to Welcome Page

Install/upgrade/Downgrade Composer

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Linux
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Installation of Composer

Use the following commands in your terminal to install Composer :

```
cd ~
curl -sS https://getcomposer.org/installer | sudo php
sudo mv composer.phar /usr/local/bin/composer
```

When moved to /usr/local/bin/composer, just type composer from any folder to run it.

Upgrade

Just type the following command to update Composer

```
composer self-update
```

Downgrade

Composer version 2.1.14 is currently the last version not impacted by the case sensitive package name issue blocking installations and updates of Mediawiki extensions. To downgrade, just use the self-update command with the version required.

```
composer self-update 2.1.14
```

After performing any self-update, with or without version specific, you can specify --rollback to go back to the previously installed version.

```
composer self-update --rollback
```

pre-release version

Finally, if you are feeling adventurous, you can update to a pre-release version by executing:

```
composer self-update --preview
```

Mediawiki - Administrator handbook

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Book
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	User Guide
Status:	Active
Access:	Micylou-restricted

load PDF

Mediawiki - Editor handbook

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Book
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	User Guide
Status:	Active
Access:	Micylou-restricted

load PDF

Mediawiki - LDAP Stack

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Integration
Last Edit By:	DochyJP

LastEdit:	2021-02-26
Document type:	User Guide
Status:	Active
Access:	Micylou-restricted

Is composed of several extensoins :

- PluggableAuth
- LDAPProvider
- LDAPAuthentication2
- LDAPUserInfo

The combination of these three extensions allos you to have logins verified to an active directory.

Do not forget in LocalSettings :
$wgGroupPermissions['user']['editmyprivateinfo'] = false; ### information is populated by the Active Directory
$wgGroupPermissions['*']['autocreateaccount'] = true; ### nécessaire pour créer le compte en local, mais la vérification se fait sur l'AD.

Seul le compte d'administrateur est pur local, ceci par sécurité.

```
##########################################################
# ACTIVE DIRECTORY LOGIN
##########################################################
#LDAP STACK
        #PluggableAuth
                wfLoadExtension( 'PluggableAuth' );
                // auto login
                $wgPluggableAuth_EnableAutoLogin = false;
                        ##>> if false, show Logout menu; if true, auto login
in session aft$
                $wgPluggableAuth_ButtonLabel='Select domain or local and
Login';
                $wgPluggableAuth_ButtonLabelMessage='Select domain or local
and Login';


        # Allow auto creation of local users if LDAP validation
                $wgGroupPermissions['*']['autocreateaccount'] = true;

        # LDAP provider
                wfLoadExtension( 'LDAPProvider' );
                // Configuration
                        $LDAPProviderDomainConfigProvider = function()
                        {
                        $config = [
```

```
                                'i-City' => [
                                        'connection' => [
                                        'server' => 'ldap001.gial.be',
                                        ###'server' => 'swdcg010.gial.be',
                                        'user' =>
'CN=SvcIntranet,OU=Deamon,OU=Fonctions and competences
(System),DC=gial,DC=be',  #di$
                                        'pass' => 'xvkghem',
                                        'options' => [
                                                        "LDAP_OPT_DEREF" => 1
                                                        ],
                                        'basedn' => 'dc=gial,dc=be',
                                        'groupbasedn' => 'dc=gial,dc=be',
                                        'userbasedn' => 'dc=gial,dc=be',
                                        'searchattribute' =>
'samaccountname',
                                        ###'searchstring' =>
'samaccountname=USER-NAME,dc=gial,dc=be',
                                        'usernameattribute' =>
'samaccountname',
                                        'realnameattribute' => 'displayname',
                                        'emailattribute' => 'mail'
                                                ]
                                        ],
                                'brucity' => [
                                        'connection' => [
                                        'server' => 'ldap001.brucity.be',
                                        ###'server' => 'swdcg010.gial.be',
                                        'user' =>
'CN=SvcIntranet,OU=Deamon,OU=Fonctions and competences
(System),DC=gial,DC=be',  #di$
                                        'pass' => 'xvkghem',
                                        'options' => [
                                                        "LDAP_OPT_DEREF" => 1
                                                        ],
                                        'basedn' => 'dc=brucity,dc=be',
                                        'groupbasedn' => 'dc=brucity,dc=be',
                                        'userbasedn' => 'dc=brucity,dc=be',
                                        'searchattribute' =>
'samaccountname',
                                        ###'searchstring' =>
'samaccountname=USER-NAME,dc=brucity,dc=be',
                                        'usernameattribute' =>
'samaccountname',
                                        'realnameattribute' => 'displayname',
                                        'emailattribute' => 'mail'
                                                ]
                                        ]
                                ];
```

```
                        return new
\MediaWiki\Extension\LDAPProvider\DomainConfigProvider\InlinePHPArray(
$config );
                        };
        # LDAP authentication
                wfLoadExtension( 'LDAPAuthentication2' );
        # Local login

                # Set to true to open a backdoor login with an internal
acount and password
                $LDAPAuthentication2AllowLocalLogin = true;

        # update of user information in the local wiki with the data
retrieved from an LDAP resource
                #wfLoadExtension( 'LDAPUserInfo' ); ### NO USER INFO
AVAILABLE AT AD GIAL ###
                #userinfo.attributes-map={"email": "mail","realname": "cn"};
```

Mediawiki Additional Namespaces

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Configuration
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	Micylou-restricted

In Localsettings.php

Remark : numbers need to be sequential.

```
## Specific language namespaces
        // Define constants for additional namespaces.
        define("NS_TEMPLATE_UK", 3000); // This MUST be even.
        define("NS_TEMPLATE_FR", 3002);
        define("NS_TEMPLATE_NL",3004);
        define("NS_TEMPLATE_UK_TALK", 3001); // This MUST be the following
odd integer.
        define("NS_TEMPLATE_FR_TALK",3003);
        define("NS_TEMPLATE_NL_TALK",3005);

        // Add namespaces.
        $wgExtraNamespaces[NS_TEMPLATE_UK] = "Template-uk";
        $wgExtraNamespaces[NS_TEMPLATE_FR] = "Template-fr";
        $wgExtraNamespaces[NS_TEMPLATE_NL] = "Template-nl";
```

```
$wgExtraNamespaces[NS_TEMPLATE_UK_TALK] = "Template-uk_talk";
$wgExtraNamespaces[NS_TEMPLATE_FR_TALK] = "Template-fr_talk";
$wgExtraNamespaces[NS_TEMPLATE_NL_TALK] = "Template-nl_talk";
```

Mediawiki Extension AdminLinks

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

Admin Links is an extension to MediaWiki that defines a special page, "Special:AdminLinks", that holds links meant to be helpful for wiki administrators; it is meant to serve as a "control panel" for the functions an administrator would typically perform in a wiki. All users can view this page; however, for those with the 'adminlinks' permission (sysops/administrators, by default), a link to the page also shows up in their "Personal URLs", between "Talk" and "Preferences".

Admin Links provides a hook and an API for other extensions to be able to add their own links and sections to the page.

You can see an example of this page with additional links and sections added by other extensions here.

MediaWiki already provides a "Special:SpecialPages" page, which lists the special pages for all the extensions installed on the wiki, which may lead you to wonder what the need is for the additional "AdminLinks" page. However, there are a number of important differences that make "AdminLinks" useful:

- "SpecialPages" holds links to every special page in the wiki, including many (such as Unused categories) that are not specifically helpful to administrators; "AdminLinks" attempts to link only to the administrator-specific ones
- "AdminLinks" can hold links to pages that are not special pages, such as the "Edit sidebar" page and documentation for various extensions
- Finally, the Admin Links extension puts a link to "AdminLinks" in administrators' personal tools, which makes things more convenient.

Installation

Download

You can download the Admin Links code, in .zip format, here.

You can also download the code directly via Git from the MediaWiki source code repository. From a command line, you can call the following:

```
git clone https://gerrit.wikimedia.org/r/mediawiki/extensions/AdminLinks.git
```

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'AdminLinks' );
```

Configuration

If you want to have the members of groups other than 'sysop' see a link to the AdminLinks page among their navigation links, you can also add lines like the following after it:

```
$wgGroupPermissions['my-group']['adminlinks'] = true;
```

Extensions using Admin Links

```
:*Cargo
:*Data Transfer
:*Maps
:*Page Forms
:*Replace Text
:*Semantic Drilldown
:*Semantic MediaWiki
:*Semantic Result Formats
```

Back to top of page - Back to Welcome Page

Mediawiki Extension ClipUpload

```
  Mediawiki Extension
```

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

```
This extension is not actively maintained but is very practical.
The ClipUpload extension is similar to the MsUpload extension with the
exception that it takes the files to be uploaded from the clipboard. It uses
"InlineAttachment" to perform its task.
Since version 1.3.0 of January 17, 2017 the browsers Chrome, Firefox and
Opera are supported.
```

Installation

Download

```
Download the extension from GIT.

sudo wget https://github.com/SLboat/ClipUpload/archive/master.zip
```

Unzip

Decompress it via

```
sudo unzip master.zip
```

Rename folder

Rename unzipped folder ClipUpload-master into ClipUpload

```
sudo mv ClipUpload-master/ ClipUpload
```

Activate the extension

Add this into the Localsettings.php with the configuration points as needed

```
require_once "$IP/extensions/ClipUpload/ClipUpload.php";
```

Configuration

This extension provides three parameters available for configuration:

- $wgClipUP_Comment
  Allows to set a standard text added to the files description upon uploading a file.
  Standard value is This file was uploaded from the clipboard ().
- $wgClipUP_MaxFileSize
  Allows to set the maximum size in KB for files uploadable by this extension.
  Standard value is 500

- $wgClipUP_CheckSameFileSize
  Allows to check if the size of a file is the same as for the previously uploaded file.
  Standard value is false

```
require_once "$IP/extensions/ClipUpload/ClipUpload.php";
// Configuration
        // $wgClipUP_Comment="" ;
                // Allows to set a standard text added to the files description upon uploading a file.
                // Standard value is This file was uploaded from the clipboard ()
        // $wgClipUP_MaxFileSize=500;
                // Allows to set the maximum size in KB for files uploadable by this extension.
                // Standard value is 500
        // $wgClipUP_CheckSameFileSize=false;
                // Allows to check if the size of a file is the same as for the previously uploaded file.
                // Standard value is false
```

Back to top of page - Back to Welcome Page

Mediawiki Extension DrawIO Editor

Mediawiki Extension

Source: DataSource

Language: English

Topic: Mediawiki

SubTopic: Mediawiki Extension

Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Download this page as PDF

Description

This is a MediaWiki extension that integrates the draw.io flow chart editor and allows inline editing of charts.

The source and full documentation is available on GitHub

Installation

Download

Clone this extension from GitHub into a folder named DrawioEditor within your wiki's extensions folder:

```
cd <your wiki root folder>/extensions
git clone https://github.com/mgeb/mediawiki-drawio-editor DrawioEditor
```

Activation

Activate the plugin in LocalSettings.php

```
require_once "$IP/extensions/DrawioEditor/DrawioEditor.php";
```

Usage

Add the following to any wiki page to insert a draw.io chart:

```
{{#drawio:ChartName}}
```

Warning

Please read these warnings carefully before use:

The actual editor functionality is loaded from draw.io. This code only provides integration.
Be aware that draw.io is an online service and while this plugin integrates the editor using an iframe and communicates with it only locally in your browser (javascript postMessage), it cannot guarantee that the code loaded from draw.io will not upload any data to foreign servers. This may be a privacy concern. Read the Privacy section in the documentation on GitHub for more information. When in doubt, don't use draw.io or this module. You have been warned!
This plugin is quite new and probably still has bugs, so it may or may not work with your installation.

Features

- draw.io chart creation and editing.
- SVG and PNG support. The file type can be configured globally and changed on a per-image basis.
- Inline Editing and javascript uploads on save, you never leave or reload the wiki page.
- Image files are transparently stored in the standard wiki file store, you don't need to worry about them.

- Versioning is provided by the file store, revert to an older version of the chart at any time.
- Draw.io original XML data is stored within the image files, so only one file must be stored per chart.
- Supports multiple charts per page.
- Supports relative and fixed chart dimensions.

Requirements

When you intend to use SVG which is recommended, you might want to install Extension:NativeSvgHandler too. Also you need a browser that supports SVG. While displaying charts may work in older browsers, especially when using PNG (SVG is default and recommended), saving charts requires a fairly recent browser.

Example

[Edit]

Back to top of page - Back to Welcome Page

Mediawiki Extension DynamicPageList

Mediawiki Extension

Source:	DataSource
Language:	English

Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The DynamicPageList extension allows wiki users to create a list of pages that are listed in a set of categories.
It was originally developed for Wikinews. It is currently installed on all language Wikinews projects (see n:Wikinews:DynamicPageList), Meta, MediaWiki.org, Wikibooks, and Wikiversity and can be installed on any small to medium sized Wikis (It is known to have scalability issues with very large wikis).
This extension was previously known as DynamicPageList/old and later Intersection. You may also be interested in the third-party DynamicPageList, which has more features.

Installation

Download

Download the extension and place the file(s) in a directory called intersection in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'intersection' );
```

Configuration

The extension also supports a couple of configuration variables. Normally you wouldn't want to change the default, but if you do you can add them to your "LocalSettings.php" after the require_once line. Here is the supported configuration variables along with their default values:

```
# Configuration variables. Warning: These use DLP instead of DPL
# for historical reasons (pretend Dynamic list of pages)
$wgDLPmaxCategories = 6;                // Maximum number of categories to
look for
$wgDLPMaxResultCount = 200;             // Maximum number of results to allow
$wgDLPAllowUnlimitedResults = false;    // Allow unlimited results
$wgDLPAllowUnlimitedCategories = false; // Allow unlimited categories
// How long to cache pages using DPL's in seconds. Default to 1 day. Set to
// false to use the normal amount of page caching (most efficient), Set to 0
to disable
// cache altogether (inefficient, but results will never be outdated)
$wgDLPMaxCacheTime = 60*60*24;          // How long to cache pages in seconds
```

Use

DynamicPageList can be used with XML-like syntax in the wiki edit view. The parameters control the pages to list, order, and display mode. See the following sections for information on each parameter. For example, the following are the oldest five pages on this wiki that were recently moved to MediaWiki.org.

```
<DynamicPageList>
category             = Pages recently transferred from Meta
count                = 5
order                = ascending
addfirstcategorydate = true
```

```
</DynamicPageList>
```

An example of a really complex DynamicPageList:

```
<DynamicPageList>
category=foo
count=20
namespace=file
addfirstcategorydate=ymd
imagewidth=70
galleryshowfilesize=yes
galleryshowfilename=yes
imagesperrow=7
gallerycaption=Galleries!
ordermethod=sortkey
order=ascending
mode=gallery
offset=12
</DynamicPageList>
```

Lists the 12th to the 34th pages in the file namespace that are also in category foo, formatting them into an image gallery, with the date they were added to category foo in the caption in year month day format, and having the images have a width of 70px

Page selection

category

category lists categories to intersect. DynamicPageList will list pages that are found in every category listed. A minimum and maximum number of entries to list is set in the extension's source code. You may use magic words like October in the category name.
The syntax is category = category name

```
<DynamicPageList>
category = Demo
category = Demo 1
</DynamicPageList>
```

notcategory

notcategory restricts the list of pages to those not in a particular category. You can use magic words like October in the category name.

```
<DynamicPageList>
category    = Demo
notcategory = Demo 1
</DynamicPageList>
```

namespace

namespace restricts the list of pages specified above to those in a particular namespace, specified by name or number (see the List of default namespaces). Any invalid name is equivalent to the main article namespace.

```
<DynamicPageList>
category  = Demo
namespace = Help
</DynamicPageList>
```

redirects

redirects determines whether or not to include redirect pages. The value can be exclude (default, don't list redirects), include (list redirects), or only (don't list pages that aren't redirects).

```
<DynamicPageList>
category  = Demo
redirects = only
</DynamicPageList>
```

stablepages

stablepages determines whether or not to include stable (flagged) pages when

using Extension:FlaggedRevisions. The value can be exclude (don't list), include (list stable and non-stable. default), or only (only list stable pages). Requires FlaggedRevs to be installed to work.

qualitypages[edit source]
qualitypages determines whether or not to include quality (flagged at the 'pristine' level) pages when using Extension:FlaggedRevisions. The value can be exclude (don't list), include (list both quality and non-quality), or only (only list quality pages).

Note: the behaviour of qualitypages changed from older versions. In older versions qualitypages=exclude automatically made stablepages=only be set. This is no longer the case.

count

count restricts the number of results that are shown. By default, the pages added to the category most recently are shown (see order). Note that the extension will not list more results than the maximum defined in the extension source code.

```
<DynamicPageList>
category = Demo
count    = 2
</DynamicPageList>
<pre>
====offset====
offset starts the list from somewhere other than the first page on the list.
For example, this can be used with count to make multi-column layouts.
<pre>
<DynamicPageList>
category = Demo
offset    = 2
</DynamicPageList>
<pre>

====Column demo====
<pre>
<table>
<tr><th colspan="3"> recent articles in
[[:category:Demo|:category:Demo]]</th></tr>
<tr><td>
```

```
<DynamicPageList>
category = Demo
count=5
</DynamicPageList>
</td>
<td>
<DynamicPageList>
category = Demo
offset    = 5
count= 5
</DynamicPageList>
</td>
<td>

<DynamicPageList>
category = Demo
offset    = 10
count=5
</DynamicPageList>
</td></tr></table>
```

Details displayed

shownamespace

shownamespace displays the namespace in the page names. The value can be true (default, display in name: Help:Contents) or false (don't display in name: Contents).

```
<DynamicPageList>
category      = Demo
shownamespace = false
</DynamicPageList>
<pre>
====addfirstcategorydate====
addfirstcategorydate shows the date each article was added to category. The value can be true (display date), false (default, don't display date), or a date specifier (ymd, md, dm, dmy, mdy, and ISO 8601). If many categories are specified, it uses the first category. The linked page names are prepended with the date formatted according to your local MediaWiki date display preferences.
<pre>
<DynamicPageList>
category             = Demo
addfirstcategorydate = true
```

```
</DynamicPageList>
<pre>
====mode====
mode determines the format of the list. The value can be:
:*unordered – bulleted list
:*ordered – numbered list
:*none – plain links with line breaks
:*gallery – image gallery, like <gallery>
:*inline – comma separated list
:*unordered:
<pre>
<DynamicPageList>
category = Demo
mode     = unordered
</DynamicPageList>
```

ordered:

```
<DynamicPageList>
category = Demo
mode     = ordered
</DynamicPageList>
```

plain links:

```
<DynamicPageList>
category = Demo
mode     = none
</DynamicPageList>
```

inline:

```
<DynamicPageList>
category = Demo
mode     = inline
</DynamicPageList>
```

mode=gallery

Gallery is a special mode where the output is an image gallery. If Extension:PageImages is installed, then the page-image if one exists will be used for non-file pages. Otherwise non-File namespace pages will just be a blank space.

```
<DynamicPageList>
category = Demo
mode     = gallery
namespace= file
</DynamicPageList>
```

Gallery mode also supports the options imagewidth (how wide the images should be in gallery, you may need to also set imageheight if using this), imageheight (how high the image should be) galleryshowfilesize (show the image size, like on category pages), galleryshowfilename (show filename like on category pages), imagesperrow (number of images per row in gallery), gallerycaption (caption for gallery).

If addfirstcategorydate is in use, the date is appended to the caption of the image.

Note, gallerycaption at the moment cannot accept links. (but it can accept things like October)

A complex gallery example might be:

```
<DynamicPageList>
count=20
namespace=file
addfirstcategorydate=ymd
imagewidth=70
galleryshowfilesize=yes
galleryshowfilename=yes
imagesperrow=7
gallerycaption=Look at my pretty gallery.
mode=gallery
shownamespace=false
</DynamicPageList>
```

suppresserrors

suppresserrors hides errors. The value can be true (hide errors) or false (default, show errors).

true:

```
<DynamicPageList>
category       = Dem
suppresserrors = true
</DynamicPageList>
```

false:

```
<DynamicPageList>
category       = Dem
suppresserrors = false
</DynamicPageList>
```

nofollow

nofollow sets rel="nofollow" on the links. This prevents spiders from getting to the page through the link from the DynamicPageList. In general this is useless, as spiders will either be able to index the page from other links, or not at all due to robots.txt/$wgDefaultRobotPolicy. In general you should not use this option unless you know what you are doing, and have a good reason. The primary usecase is that on the English Wikinews, Google News considers anything linked from the main page to be an article.

```
<DynamicPageList>
category       = Demo
nofollow       = true
</DynamicPageList>
```

googlehack

This is a hack for wikinews. Previously, Google News only considered articles with numbers in them to be news articles [1], so this appends ?dpl_id=<id of article> to the end of the urls on the list. You shouldn't use this option unless you have a good reason to (however it won't hurt anything). This replaces the showcurid option that is no longer supported.

```
<DynamicPageList>
category       = Demo
googlehack       = true
</DynamicPageList>
```

Order

ordermethod[edit source]
ordermethod determines the order and date to display. The value can be:

- categoryadd – default, sort by date pages were categorised for first category
- lastedit – sort by date pages were last edited
- length – by article length
- created – by article creation date
- categorysortkey – essentially alphabetical order by the sort key of the first category
- sortkey – alias of categorysortkey

It should be noted, that lastedit really sorts by the last time the page was touched. In some cases this is not equivalent to the last edit (for example, this includes permission changes, creation or deletion of linked pages, and alteration of contained templates). If you do not specify a category, some of these order methods will not work—specifically, categoryadd and categorysortkey fallback to created if there is no category clause.

```
<DynamicPageList>
category             = Demo
addfirstcategorydate = true
ordermethod          = lastedit
</DynamicPageList>
```

alphabetical example

```
<DynamicPageList>
category             = Demo
ordermethod          = sortkey
</DynamicPageList>
```

order

order decides the sort direction. The value can be descending (default, most recent to oldest) or ascending (oldest to most recent).

```
<DynamicPageList>
category  = Demo
order     = ascending
</DynamicPageList>
<+pre>
====Templates====
To use inside a template and pass parameters, use parser functions etc., use
the {{#tag: function:
<pre>
{{#tag:DynamicPageList|
category = {{{1}}}
count    = {{#if: {{{2|}}} | 10 | 20 }}
ordermethod = sortkey
order = ascending
}}
```

The template page itself will display an error message because the category value is empty on the template page unless you put the template inside tags. Example:

> Error: You need to include at least one category, or specify a namespace!

Back to top of page - Back to Welcome Page

Mediawiki Extension EmbedVideo

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The EmbedVideo extension adds a parser function called #ev for embedding video clips from over 24 popular video sharing services in multiple languages and countries. It also adds video and audio media handlers to support transforming standard [[File:Example.mp4]] file links into embedded HTML5 <video> and <audio> tags.

Installation

Download

Download and place the file(s) in a directory called EmbedVideo in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'EmbedVideo' );
```

Configuration

Variable	Default Value	Description
$wgEmbedVideoAddFileExtensions	TRUE	Boolean - Enable or disable adding video/audio file extensions to the list of allowable files to be uploaded.
$wgEmbedVideoEnableVideoHandler	TRUE	Boolean - Enable or disable the video media handlers for displaying embedded video in articles.
$wgEmbedVideoEnableAudioHandler	TRUE	Boolean - Enable or disable the audio media handlers for displaying embedded audio in articles.
$wgEmbedVideoDefaultWidth		Integer - Globally override the default width of video players. When not set this uses the video service's default width which is typically 640 pixels.
$wgEmbedVideoMinWidth		Integer - Minimum width of video players. Widths specified below this value will be automatically bounded to it.
$wgEmbedVideoMaxWidth		Integer - Maximum width of video players. Widths specified above this value will be automatically bounded to it.
$wgFFmpegLocation	/usr/bin/ffmpeg	String - Set the location of the ffmpeg binary.
$wgFFprobeLocation	/usr/bin/ffprobe	String - Set the location of the ffprobe binary.

Developper note

The major benefit of using a parser function as the injection mechanism for video clips (as opposed to an extension tag) is that parser functions can utilize template parameters (of the form {{{1}}}).

For example, say you found yourself making a lot of YouTube videos in thumbnail size, right-aligned like an image. You might make a template called Template:Youtube_tn which could contain this:

```
<div class="thumb tright">
{{#ev:youtube|{{{1}}}|{{{2|100}}}}}
</div>
```

And then call it like this:

```
{{youtube tn|someYoutubeId}}
```

Back to top of page - Back to Welcome Page

Mediawiki Extension Flex Diagrams

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension

Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

Flex Diagrams is an extension that lets you create, and display, diagrams using three different JavaScript libraries: bpmn-js (for BPMN diagrams), DHTMLX Gantt (for Gantt charts), and Mermaid (for various kinds of diagrams, using the Mermaid JavaScript library). Each of these three gets its own namespace.

Any such diagram page will get, in addition to the standard "Edit" tab, an "Edit diagram" tab that allows for editing these diagrams in a graphical way. Thus, any diagram can be edited either as text, or graphically. (Although the Mermaid editor is unfortunately not a true graphical editor, but simply a text editor that includes a preview element.)

Installation

Download

You can also download the code directly via Git from the MediaWiki source code repository (see available tags). From a command line, call the following:

```
git clone
https://gerrit.wikimedia.org/r/mediawiki/extensions/FlexDiagrams.git
```

Activation

Once you have obtained a "FlexDiagrams" directory and placed it in your /extensions directory, add the following to LocalSettings.php:

```
wfLoadExtension( 'FlexDiagrams' );
```

Configuration

Usage

Creating diagrams

A BPMN, Gantt, or Mermaid diagram can be created by creating a page in the appropriate namespace (BPMN:, Gantt:, or Mermaid: respectively), for example:

```
BPMN:Approval process
```

Once the page is created (or while editing in either "Source" or "Visual Editor") a new edit tab called "Edit diagram" (or "Create diagram" if the page has not been saved yet) will be shown, clicking on this editor launches an interactive diagram editor as shown below.

Adding links to BPMN diagrams

One customization specific to the Flex Diagrams extension is that, when generating a BPMN diagram, you can make any element be a link to a different wiki page. To do this, add the name of the page you want to link to, within double square brackets, at the end of the element title - after either a space or a newline. An example would be a node called "Get approval". Let's say there's a separate page, "BPMN:Approval process", which details all the steps of the process referred to as "Get approval". We can make the "Get approval" node link to "BPMN:Approval process" by changing its label to read "Get approval [BPMN:Approval process]". When the diagram is displayed, the link part will not be shown on the screen; instead, the entire node will appear as

blue and become a clickable link. You can see examples of how such linking gets displayed in the gallery below.

Displaying diagrams

Any diagram can be displayed on any page by using the #display_diagram function. This function takes in a single argument, a page name, and displays the diagram defined on that page. An example would be:

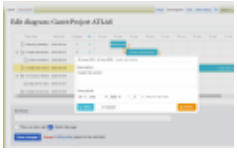
{{#display_diagram:BPMN:Approval process}}

Screenshots

-

-

-



-

See also

Extension:Cognitive Process Designer

[Back to top of this page]

[Back to Welcome Page]

Mediawiki Extension Iframe

Mediawiki Extension

Source: DataSource

Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Copyright & security issues

Embedding external webpage into MediaWiki has two issues:

- Copyright violation: Embedding an external web page may violate copyrights. So, make sure that the domains that can be used in the extension are trusted!
- Security issue: External web pages may contain malicious code. Again make sure that the domains that can be used in the extension are trusted!

The extension uses keys to select web servers which are given by the wiki adminstrator during installation. Therefore, behind the key should only be trusted domains.

Known problems

The extension does not work properly with the MobileFrontend extension.

Installation

Clone the GIT repository

Go to the extensions directory and clone it from GitHub

```
    git clone https://github.com/sigbertklinke/Iframe
```

Add to the bottom of LocalSettings.php

```
wfLoadExtension( 'Iframe' );
```

The global variable $wgIframe controls various aspects of the extensions, e.g. additional web adresses, delay of iframe loadings etc.

Adding other web addresses

If you want to add own keys/servers then add in LocalSettings.php after loading the extension

```
 $wgIframe['server']['mykey'] = [ 'scheme' => 'https', 'domain' => 'mydomain' ];
```

This will allow you to use in wiki pages:

```
<iframe key="mykey" path="" />
```

Category

For easy finding of all wiki pages which have a <iframe ... /> inside, a category is added by

.

The name of the category can be controlled in LocalSettings.php by setting

```
$wgIframe['category'] = 'Iframe';
```

Notes:

The extension does NOT create the page Category:Iframe, that is left to the wiki administrator or user.
If you create the page Category:Iframe then it might be useful to use hidden categories.

Width and height of an iframe

The default width and height of an iframe is 800x600 pixel. This can be set in LocalSettings.php via

```
$wgIframe['width'] = 640;
$wgIframe['height'] = 400;
```

If no width and height is given then these defaults are used.

Delay of iframe loading

Usually a HTML document is shown if all parts of a document are loaded. If a iframe on a server, e.g. a Shiny server, triggers a lengthy computation then

the document will appear only after all iframe(s) are loaded. Therefore a delay mechanism with JavaScript is included which starts the loading of the iframe(s) only after the document is ready. By $wgIframe['delay'] the delay is controlled:

If in LocalSettings.php

```
$wgIframe['delay'] = -1;
```

(or any negative value) is set then the iframe(s) will be loaded immediately, no delay will take place.

If in LocalSettings.php

```
$wgIframe['delay'] = 100;
```

(the default) is set then the first iframe will be loaded 100 ms after the document is loaded, the second 200 ms, and so on.

Mediawiki Extension MassEditRegex

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation

Status: Active

Access: free

Description

The MassEditRegex extension allows administrators perform a single edit across multiple pages in one step, by running a regular expression over the content of each page. This is well suited to performing simple edits such as renaming a template, adding pages to a category, or correcting typos (all of which can be done in the same edit operation by supplying multiple regular expressions.)

Installation

Download

Download and place the file(s) in a directory called MassEditRegex in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
require_once "$IP/extensions/MassEditRegex/MassEditRegex.php";
```

Configuration

This extension comes with an extra user right called "masseditregex" to allow

fine grained control of which user groups may use this extension. By default it is assigned to no user group. In case you would like to assign it to a user group e.g. "masseditregexeditor", add the following code to you "LocalSettings.php" file right after the lines added in step (3) of the installation process:

```
$wgGroupPermissions['masseditregexeditor']['masseditregex'] = true;
```

Alternatively you may grant the permission to an existing user group, e.g. "sysop" by adding the following line:

```
$wgGroupPermissions['sysop']['masseditregex'] = true;
```

Known issue

Server-side execution is the default as it (should) be faster. However editing large numbers of pages will cause server timeouts due to PHP limits on script execution time. To work around this (and perhaps to get better feedback on progress) you can tick 'execute in browser' to have the job tracking done on the client-side. This should mean timeouts are no longer a problem.

Usage

Once the extension has been installed and permission masseditregex to use MassEditRegex has been given, go to "Special:MassEditRegex" and fill out the edit form. Familiarity with regular expressions is required to use this extension.

See also

Extension:Replace Text - A similar extension which however will not work if you set $wgCompressRevisions = true; for your wiki.

Back to top of page - Back to Welcome Page

Mediawiki Extension MsInsert

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The MsInsert extension adds a dropdown menu to the editor (both the WikiEditor and the regular editor), that allows you to insert into the wikitext the content of any wiki page listed in the dropdown menu. The wiki pages are added to the dropdown menu from LocalSettings.php

Installation

Download

Download and place the file(s) in a directory called MsInsert in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'MsInsert' );
```

Configuration

To add a page to the dropdown menu, add it to the $wgTemplates array in your LocalSettings.php, like so:

```
$wgTemplates = [ 'Template:Test', 'Main Page', 'Talk:Main Page' ];
```

See also

Extension:MsLinks
Extension:MsUpload
Extension:MsCatSelect
Extension:MsWikiEditor

Back to top of page - Back to Welcome Page

Mediawiki Extension MsUpload

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The MsUpload extension allows users to upload multiple files by drag & drop in the standard MediaWiki edit page.

Features

- Easily upload one or multiple files simultaneously.
- File upload integrated in the edit page.

- Upload via drag and drop (if HTML5 is available in browser).
- Add the file into the edit page as a link.
- Add the files into the edit page as a gallery.
- Change the name of the file before uploading.
- Add files to the current category, if the page being edited is a category page.

Installation

Download

Download and place the file(s) in a directory called MsUpload in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'MsUpload' );
```

Configuration

Features

Below are the special extension features that you can set in your LocalSettings.php:

- $wgMSU_useDragDrop = true; // Should the drag & drop area be shown?

(Not set by default)

- $wgMSU_showAutoCat = true; // Files uploaded while editing a category page will be added to that category
- $wgMSU_checkAutoCat = true; // Whether the checkbox for adding a category to a page is checked by default
- $wgMSU_useMsLinks = false; // Insert links in Extension:MsLinks style?
- $wgMSU_confirmReplace = true; // Show the "Replace file?" checkbox
- $wgMSU_imgParams = 'thumb|none'; // Any image parameter(s), delimited by pipes, e.g., {width}px, left, right, center, none, border, frameless, frame, thumb etc.

Relevant global configuration variables

Below are some of the relevant global configuration variables that you can set in your LocalSettings.php:

- $wgEnableWriteAPI = true; // Enable the API
- $wgEnableUploads = true; // Enable uploads
- $wgAllowJavaUploads = true; // Solves problem with Office 2007 and newer files (docx, xlsx, etc.)
- $wgGroupPermissions['user']['upload'] = true; // Allow regular users to upload files
- $wgFileExtensions = array('png','gif','jpg','jpeg','doc','xls','pdf','ppt','tiff','bmp','docx','xlsx','pptx');
  // Make sure that the file types you want to upload are allowed:

Remark

You should know

- Without HTML5 support Adobe Flash is used
- The maximum file size depends on MediaWiki settings but a lower limit may be set.
- Uses the plupload upload handler, which comes included with the extension
- Adds an upload icon to the editor and WikiEditor.

See also

Extension:MsWikiEditor
Extension:MsUpload
Extension:MsLinks
Extension:MsInsert

Back to top of page - Back to Welcome Page

Mediawiki Extension MultiLanguageManager

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki

SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The MultiLanguageManager extension allows you to add multi-languages capacities to a wiki:

- You can define a list of available languages
- You can define a page to be a variant of another page within the same wiki
- On a page with variants a special navigation will be displayed ("flags") that allow to switch between those pages
- The interface language for anonymous users will automatically be changed to match the language of the page

ATTENTION : This extension does not use the built-in Page content language mechanism of MediaWiki.

Installation

Download

Download and place the file(s) in a directory called MultiLanguageManager in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'MultiLanguageManager' );
```

Run the update script which will automatically create the necessary database tables that this extension needs.

Configuration

Available languages

The configuration is done using the $mgAvailableLanguage variable which contains the code of language and the code of its name in the file MultiLanguageManager_setup.php.

```
$mgAvailableLanguage = array(
   'en' => 'multilanguagemanager_en',
   'fr' => 'multilanguagemanager_fr'
);
```

User rights

You can manage who has the rights to modify the linguistic parameters of the pages. You can set those users rights to match with an existing right using the $mgLanguagePermisionsKey variable . For example to set the linguistic rights to match the edition article rights:

```
$mgLanguagePermisionsKey = 'edit';
```

But you could also choose an other rights : language :

```
$mgLanguagePermisionsKey = 'language';
```

Then think of set the rights according to the groups:

```
$wgGroupPermissions['*'][$mgLanguagePermisionsKey] = false;
$wgGroupPermissions['user'][$mgLanguagePermisionsKey] = true;
$wgGroupPermissions['sysop'][$mgLanguagePermisionsKey] = true;
```

Back to top of page - Back to Welcome Page

Mediawiki Extension NativeSvgHandler

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension

Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The NativeSvgHandler extension allows SVG files to be served directly to clients for client-side rendering.

Installation

Download

Download and place the file(s) in a directory called NativeSvgHandler in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'NativeSvgHandler' );
```

Security

NativeSvgHandler serves SVG directly to clients through image tags, which is not vulnerable to XSS attack.

XSS attacks are only relevant when it is accessed directly through original file on the file page, which is neither in the scope nor in control of this extension.

Back to top of page - Back to Welcome Page

Mediawiki Extension PDF Embed

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The PDFEmbed extension allows PDF files (uploaded to your wiki) to be embedded into a wiki page using <pdf></pdf> tags. The PDF file extension is automatically added and necessarily default permissions are configured. Future functionality will allow this extension to act as a media handler for PDF files.

Installation

Download

Download and place the file(s) in a directory called PDFEmbed in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'PDFEmbed' );
```

Configuration

If the default configuration needs to be altered add these settings to the "LocalSettings.php" file below the require:

```
// Default width for the PDF object container.
$wgPdfEmbed['width'] = 800;
```

```
// Default height for the PDF object container.
$wgPdfEmbed['height'] = 1090;
```

Usage

The PDF to be embedded must first be uploaded to the wiki through "Special:Upload". The namespace prefixed file name is then used between the tags to embed it in a page.

```
<pdf>File:Example.pdf</pdf>
```

With optional width and height.

```
<pdf width="500" height="300">File:Example.pdf</pdf>
```

Scroll to page 10 after document is rendered.

```
<pdf page="10">File:Example.pdf</pdf>
```

See also

Extension:PdfHandler Extension:Widgets (PDF-Widget)

Back to top of page - Back to Welcome Page

Mediawiki Extension PageForms

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

Page Forms (known before 2016 as Semantic Forms) is an extension to MediaWiki that allows users to add, edit and query data using forms. It was originally created as an offshoot of the Semantic MediaWiki extension, to be able to edit templates that store their parameters via SMW, which is why it was originally called "Semantic Forms". However, it can now also work with the alternative Cargo extension, or with neither extension installed.

Very simply, Page Forms allows you to have forms for creating and editing pages on your wiki, as well as for querying data, all without any programming. Forms can be created and edited not just by administrators, but by users themselves.

The main components of Page Forms functionality are form definition pages, which exist in a separate namespace, 'Form:'. These are pages consisting of markup code which gets parsed when a user goes to a form. Since forms are defined strictly through these definition pages, users can themselves create and modify forms, without the need for any actual programming.

The Page Forms extension is mostly used to add and edit calls to infobox-style templates within pages. A form allows a user to populate a pre-defined set of templates, as well as page sections, for any page; if Cargo or Semantic MediaWiki are used, the data within the templates can then be stored and queried.

Forms can also be used to edit the data in an existing page, and you can enable an "edit with form" tab to show up on any page.

Page Forms also supports autocompletion of fields, so users can easily see what the previously-entered values were for a given field. This greatly helps to avoid issues of naming ambiguity, spelling, etc.

Data in a page that doesn't fit into the form, like a free-form text description of the page's subject, isn't ignored when the page is edited with a form; rather, it is placed into a separate input box called "free text".

Page Forms also includes a wide variety of other functionality related to forms.

Below are links to the entire documentation.

Installation

Download

You can download] the Page Forms code from github, in .zip format or clone it usint the command

```
git clone https://gerrit.wikimedia.org/r/mediawiki/extensions/PageForms.git
```

Activation

Once you have downloaded the code as a directory called PageForms in your extensions folder, you just need to add the following line to your LocalSettings.php file:

```
wfLoadExtension( 'PageForms' );
```

Configuration

There are four User rights defined for Page Forms:

1. 'createclass' is the permission to create new classes (see Extension:Page Forms/Special pages). By default it is given to all members of the 'user' group.
2. 'editrestrictedfields' is the "permission" to edit restricted fields. By default it is given to all members of the 'sysop' group.

3. 'multipageedit' is the "permission" to use the MultiPageEdit special page, which allows to edit multiple pages through a single spreadsheet-like interface. By default it is given to all members of the 'user' group.
4. 'viewedittab' is the "permission" to see the "edit with form" tab. By default it is given to everyone (i.e, '*').

You can modify the set of users who have any of these permissions. For example, to have the "edit with form" tab show up only for administrators, you could add the following, below the inclusion of Page Forms:

```
$wgGroupPermissions['*']['viewedittab'] = false;
$wgGroupPermissions['sysop']['viewedittab'] = true;
```

Quick start guide

So, you've set up all the software on your site. Now what? Now it's time to create the structures to hold, display and enable adding and editing of your data; thankfully all of this can be done simply by creating various wiki pages.

The first step should be to figure out your data structure. What types of pages will the site have? What data will be contained on each one? You can change all of this around later, but you should have a starting plan. Once you have that (ideally, written down), you can move on to creating the necessary wiki pages. You will need to create or modify pages for templates and forms, probably categories as well, and (if you're using Semantic MediaWiki) properties.

You can create all of these from scratch, although helper forms also exist to help with creating them. The right approach depends on which other extensions you're using; we will limit this to Semantic Mediawiki.

The easy way - Special:CreateClass

The page Special:CreateClass, defined by Page Forms, provides a single helper form to define everything about a data "class" in one place. If you fill out the fields and hit "submit", the page will create all the necessary property, template, form and category pages automatically. Then you can go to the form page that was created, and start entering data.

If not all the pages seem to be getting created, it could be that your job queue lacks enough memory. See the job-queue-related item here for how to add more memory.

The more comprehensive way[edit] You should take the following steps:

- Create properties. The basic building blocks of any semantic site are the connections between data, which in Semantic MediaWiki are known as properties. A property is used to specify a single piece of information about the topic of this page. Every property should be defined on your wiki, with a page in the "Property:" namespace. Create each property by going to the Special:CreateProperty page.
- Create templates. A template sets the display of the data on a page, holds the markup to turn the data into actual semantic information, and (often) defines the page as being of a certain category, and thus of a certain page type. There will generally be one template per page type, although sometimes a single page type will contain more than one

template. A template can also be used for aggregation, by displaying a list of all other pages that have a certain relationship to this page (see the inline queries documentation for much more information on this). Create the template(s) by going to the Special:CreateTemplate page.
- Create forms. Now you can create forms to allow users to easily add and edit pages of various types. There should be one form per page type; the form will populate the template or templates that this page type contains. As before, there's a special page to create new forms: Special:CreateForm.
- Create categories. One template in each form should tag each article that the form creates as being part of a certain category. You should create a page for each such category, and specify a default form for it so that every article is automatically editable with the same form that created it. The easiest way to do that is by using the Special:CreateCategory page.
- Enable links to forms. Besides creating categories that have a default form, there are other steps you should take to enable users to access the forms that you have created. These include adding links to add data in the sidebar and elsewhere, and using #formredlink in order to have red-links to nonexistent pages point to a form for creating them. These actions are all explained in the section "Linking to forms".
- Add links on sidebar and elsewhere. The sidebar (which, for English-language wikis, can be edited at the page "MediaWiki:Sidebar") should hold links to add each of the data types, and to the categories for each of these data types as well. You can also include such links on the main page and elsewhere.
- Add data. Now you can start entering data! Just go to the form page(s), and start creating pages.
- Customize. Once the structure is in place, you can customize all you want - changing the look-and-feel of the entire wiki, of the various templates, and of the forms, adding and changing fields and adding and changing inline queries.

Example

Example

Let's go through a specific example of setting up a Page Forms-based wiki from scratch, in the "more comprehensive way".</translate> <translate> Let's say that you want to create a wiki that shows all the books you have at home.</translate> <translate> We'll then go through the previous steps specified, with some parts different for Cargo vs. SMW.

1. Figure out the data structure.</translate> <translate> You want two kinds of pages on your wiki: one for books, and one for authors.</translate> <translate> Each book page should hold a title, an author name (or names), a genre (or genres), a year of publication, and a number of pages.</translate> <translate> Each author page should hold the author name, their country of origin, and a list of all the books they have written that you own.

2. Create properties.</translate>

<translate> With Cargo</translate>	**<translate> With SMW</translate>**

<translate> There is no equivalent step for Cargo.</translate>	<translate> For books, there is no need to create a property for the title, since the title of each page representing a book will be the title of that book.</translate> <translate> That leaves four fields:</translate> <translate> • Author(s) - using the "Create a property" page (at `Special:CreateProperty`), you create a property called **"Was written by"**, of type "<tvar|page>Page</>" (since each author will have their own page).</translate> <translate> • Genre - you create a property called **"Has genre"**, of type "<tvar|txt>Text</>", and give it allowed values such as: "Art, Cookbook, Fiction, History, Poetry, Science".</translate> <translate> • Year of publication - you create a property called **"Was published in year"**, of type "<tvar|nb>Number</>".</translate> <translate> • Number of pages - you create a property called **"Has number of pages"**, of type "<tvar|nb>Number</>".</translate> <translate> For author pages, you only need to create one more property:</translate> <translate> • Country of origin - you create a property called **"Is from country"**, of type "<tvar|txt>Text</>".</translate>

<translate> **3. Create templates.**</translate> <translate> You need to create two templates, one for books and one for authors, both at the "Create a template" page at `Special:CreateTemplate`.

First, books:</translate>

<translate> With Cargo</translate>	**<translate> With SMW</translate>**

create>Special:CreateTemplate</>, set the template name to be "Book", and set the category name to be "Books".</translate> <translate> You add to it four fields, and for each field you specify at least three values: its field name in the template source code, its display label on each page, and its type.</translate> <translate> There are a few other values you can set, but they will not be necessary for most of these fields.</translate>
<translate>
1. For the first field, set the field name to be "Authors", the label to be "Author(s)" and the type to be "<tvar|page>Page</>".</translate> <translate> You also need to check the box indicating that this field can hold a list of values (since a book can have more than one author).</translate>
<translate>
1. Click **Add field** and then, for the second field, the three values to be "Genres", "Genre(s)" and "<tvar|str>String</>".</translate> <translate> You again need to check the box indicating that this field can hold a list of values.</translate>
<translate> You should then specify the set of allowed values; this can be done as either a simple list, or a hierarchy.</translate> <translate> You could set a simple list like "Art, Cookbook, Fiction, History, Poetry, Science", and so on; or you could set a complete hierarchy, which could look like:</translate>

```
* Fiction
** Science fiction
*** Cyberpunk
** Historical fiction
```

<translate> ...and so on.</translate>
3. <translate> Click **Add field** and then, for the third field, set the values to be "Year", "Year of publication" and "Date" (a better choice than "<tvar|int>Integer</>").</translate>
4. <translate> Click **Add field** and then, for the fourth field, set them to be "Number of pages", "Number of pages" and "Integer".</translate>
5. <translate> Click the **Save page** button.</translate>

create>Special:CreateTemplate</>, set the template name to be "Book", and set the category name to be "Books".</translate> <translate> You add to it four fields, and for each field you specify three values: its field name in the template source code, its display label on each page, and the semantic property it populates.</translate>
<translate>
1. For the first field, set the field name to be "Authors", the label to be "Author(s)" and the semantic property to be "Was written by", and check the box indicating that this field can hold a list of values (since a book can have more than one author);</translate>
<translate>
1. Click **Add field** and then, for the second field, set the three values to be "Genres", "Genre(s)" and "Has genre", and also check the box for a list of values;</translate>
<translate>
1. Click **Add field** and then, for the third field, set the values to be "Year", "Year of publication" and "Was published in year";</translate>
<translate>
1. Click **Add field** and then, for the fourth field, set the values to be "Number of pages", "Number of pages" and "Has number of pages".</translate>
<translate>
1. Click the **Create** button.</translate>

<translate> Next, create the template for Authors - using the "Create a template" page, you create a template called **"Author"**, and set it to define the category "Authors".</translate>

<translate> With Cargo</translate>

<translate> With SMW</translate>

1. <translate> Just add to this template one field, for which you set the three values to be "Country", "Country of origin" and "<tvar|page>Page</>".</translate>
2. <translate> Click the **Save page** button.</translate>

1. <translate> Just add to this template one field, for which you set the three values to be "Country", "Country of origin" and "Is from country".</translate>
2. <translate> For this template, you also fill out the "Aggregation" fields in the form, since you want each author page to list all the books the author has written.</translate> <translate> Under "Semantic property", you select "Was written by", and for "Title for list" you put "Books by this author".</translate>
3. <translate> Click the **Save page** button.</translate>

<translate> **4. Create forms.**</translate> <translate> You need to create two forms, one for books and one for authors:</translate>

- <translate> Books - using the "Create a form" page (at <tvar|create>`Special:CreateForm`</>), you make a form called **"Book"**, select the template "Book" and click the **Add** button.</translate> <translate> You could also optionally add a section for the plot of the book - to do that, add a section (not a template) called "Plot".</translate> <translate> You do not do any further customizations on this form, but just save the form once you have created it.</translate>
- <translate> Authors - using the "Create a form" page, you make a form called **"Author"**, and add to it the template "Author".</translate> <translate>

As before, you do not do any further customizations - you just save the page.

5. Create categories.</translate> <translate> You need to create two categories, one for books and one for authors; you have already specified these categories' names before:</translate>

- <translate> Books - using the "Create a category" page (at <tvar|1>`Special:CreateCategory`</>), you make a category called **"Books"**, and give it the default form "Book".</translate>
- <translate>

Authors - using the "Create a category" page, you make a category called **"Authors"**, and give it the default form "Author".

6. Enable links to forms.</translate>

<translate> With Cargo</translate>	<translate> With SMW</translate>
tpl>`Template:Book`</>) find the portion of the display containing the tag:</translate> `{{{Authors|}}}` <translate> Replace this with something like:</translate> `{{#arraymap:{{{Authors|}}}|,|x|{{#formredlink:target=x|form=Author}}}}`	tpl>`Template:Book`</>) most likely has a portion that looks like</translate> `{{#arraymap:{{{Authors|}}}|,|x|[[Was written by::x]]}}` <translate> Replace this with something like</translate> `{{#arraymap:{{{Authors|}}}|,|x|{{#formredlink:target=x|form=Author}}{{#set:Was written by=x}}}}`

<translate> This will make links to author pages that don't exist yet point to the form for creating those pages.

7. Add data.</translate> <translate> Using the new "Book" form (at <tvar|1>`Form:Book`</>), you create pages for some books; you check to make sure that both the form and the resulting pages look alright.</translate> <translate> Each book page will now have a red link to its author, that should point to the "Author" form for adding that author (you may need to refresh the page for the right link to show up).</translate> <translate> You can then add a few authors, either through

these links or from the form page directly, to make sure that this setup works correctly too.</translate> <translate> Each created author's page should show a list of the books you've saved that have that person as the author.

8. Add links on sidebar and elsewhere.</translate> <translate> Your structures are now all set up, and you can add links to the sidebar for easy adding and viewing, by modifying the page <tvar|bar>`MediaWiki:Sidebar`</> (or its equivalent if you're using a language other than English).</translate> <translate> You can add links to the "Book" and "Author" forms to make it easy to add new ones.</translate> <translate> You could also link to <tvar|drilldown>Special:Drilldown</> (if you have Cargo installed), or <tvar|BrowseData>Special:BrowseData</> (if you have the [[<tvar|1>Extension:Semantic Drilldown</>|Semantic Drilldown]] extension installed), to let users view all the data at a glance.

Using Page Schemas

The Page Schemas extension is specifically intended for the automated creation of data structures, including forms. Its big advantage over the standard PF helper forms is that it can be used to create and modify these data structure pages many times, not just create them one time. It can be used with either Cargo or Semantic MediaWiki installed, or with neither one installed. If Cargo and/or SMW are installed, the Page Schemas form will include many additional form fields in order to create the additional tags needed. To create a page schema, you need to go to the category page (whether or not it has been created yet), go to the tab action for creating that page's schema, fill out the form, then go the tab action for generating pages based on that schema. See the Page Schemas documentation for more information.

Back to top of page - Back to Welcome Page

Mediawiki Extension PageSchema

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

Page Schemas is a MediaWiki extension that allows for defining all the schema information about a "class", or data type - for naming, display, data input, and browsing - within a single piece of XML contained within a category page. This XML then is used to generate all the necessary additional

pages needed to put the schema into effect, such as template pages and, for instance, if Page Forms is installed, form pages. The XML can be created and edited via a helper form, preventing users from having to generate or modify it manually.

Page Schemas allows other extensions to define their own fields to add to the XML, using hooks. Currently, the following extensions hook into Page Schemas:

- Cargo
- Page Forms
- Semantic MediaWiki
- Semantic Drilldown
- Semantic Internal Objects

Installation

Download

You can download the Page Schemas code, in .zip format, here.

You can also download the code via Git from the MediaWiki source code repository. From a command line, you can call the following:

```
git clone https://gerrit.wikimedia.org/r/mediawiki/extensions/PageSchemas.git
```

Activation

After you've obtained a 'PageSchemas' directory, place this directory within the main MediaWiki 'extensions' directory. Then, in the file 'LocalSettings.php' in the main MediaWiki directory, add either one of the following lines:

```
wfLoadExtension( "PageSchemas" );
```

or

```
include_once( "$IP/extensions/PageSchemas/PageSchemas.php" );
```

Configuration

Page Schemas defines two pages that can show up as tabs on category pages: "Edit schema" (which appears as "Create schema" if no schema exists yet) and "Generate pages". "Edit schema" allows for creating or editing the XML via the helper form, while "Generate pages" provides a button that lets the administrator create the template(s), semantic properties, form etc. automatically from the XML definition. "Generate pages" also includes checkboxes for each of the pages to be generated, to let administrators choose not to override certain wiki pages that already exist.

Any user who is allowed to edit a certain category page will see the "Edit schema/Create schema" tab for that category, and will be allowed to use that page to modify the schema.

In order to generate pages, a user must have the 'generatepages' permission. By default, this

permission is given only to members of the 'sysop' user group. You can add to or modify the set of allowed users, though, by modifying the $wgGroupPermissions array in LocalSettings.php. To enable all registered users to generate pages from the schema, for instance, you would add the following:

```
$wgGroupPermissions['user']['generatepages'] = true;
```

Tutorial video

You can see a screencast video for Page Schemas created by Yury Katkov, explaining how it's used and why it can be useful.

Sample XML structure

Here is some sample XML, to be contained within a page called "Category:Cities", used to define the schema for pages within that category. This section defines the name of the form, and information for a single field of that class, "Population".

'"`UNIQ--PageSchema-00000062-QINU`"'

Back to top of page - Back to Welcome Page

Mediawiki Extension Quiz

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

Installation

Download

Download and place the file(s) in a directory called Quiz in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'Quiz' );
```

Create your Quiz

Adding a new question type

Quiz was conceived to facilitate the addition of new question types. If the basic types (multiple choice with single/multiple responses) are not enough for you, you have the possibility of easily creating and integrating the type of questions which you need. For that you must have some knowledge of PHP.

- The first thing that you must do is choose a syntax (the simplest possible) for your question type. The restrictions are:

1. The syntax of the question's header is fixed. The question has to be placed between curly brackets - i.e. { and }
2. For consistency, it is recommended to use || to mark the beginning of the feedback section ("correction").

- Choose a new code name for your question type (example: "questionTypeName") as well as a symbol (currently the symbol "()" is used for single response multiple choice because it looks like a radio button, and the symbol "[]" is used for multiple response multiple choice because it looks like a checkbox).
- Add a "case" at the beginning of the parseParameters's "switch", as follows :

```
case 'symbol':
```

<pre> $this->mType = "questionTypeName"; break;

- Add a parser function to Quiz.php.
  This function must convert a question object from quiz syntax to HTML and set a state to the question (right, wrong, error, NA or unmarked), according to syntax errors and eventually posted answers. Name the function "questionTypeNameParseObject" and implement as follows :

```
 /**
 * Convert a basic type object from quiz syntax to HTML.
 *
 * @param  $input                                A question object in quiz
syntax
 *
 * @return $output                               A question object in HTML.
 */
 function questionTypeNameParseObject($input)
```

```
{
// your code here.
$this->setState(""); // Set the state if needed.
return $output;
}
```

1. Write down your syntax specifications which will be added to the help. Please read first the existing help.
2. Once this is all done, contact me to include officially your question type inside the extension.

Back to top of page - Back to Welcome Page

Quiz documentation

This is the help page for the quiz extension.

The quiz extension consists of very simple markup. This page is long because of the many possibilities which the quiz extension opens up. For a simpler and colourful guide to quizzes suitable for beginners, try the simple quiz help page.

Overview

Version

- Version 1.0 of the quiz extension is now (20 May 2007) active on Wikiversity.
- This documentation covers all the major features and most of the minor ones.
- A few exotic features are not yet covered, such as shuffle control using {X} {!X} {/X} tags.

Pedagogical considerations

- Where are the strengths and weaknesses of Wikiversity quizzing?
- How can you best use the quiz extension as a teaching tool?
- Wikiversity compared to Moodle - offers a detailed consideration of the quiz feature differences between Moodle and Wikiversity, including suggestions as to how each is best used.

How to create a quiz

The quiz tag

wikitext (what you write)	rendering (what you see)	comments

<quiz> *insert questions here* </quiz>	Point added for a correct answer: 1 Points for an incorrect answer: 0 Ignore the questions' coefficients: ☐ Shuffle questions 1 Sample Question 1 ○ The correct answer. ○ Distractor. ○ Distractor. ○ Distractor. 2 Sample Question 2 ○ The correct answer. ○ Distractor. ○ Distractor. ○ Distractor. Submit	The *quiz* tags do the following: • They insert some standard quiz options at the top and a colour-key for marking. The colour-key only appears when the question is marked. • They add a "submit" button at the bottom for when the learner has finished the quiz and wants it marked. • Everything between the quiz tags is parsed by the quiz extension and turned into questions if possible. Theoretically it is possible to put *quiz* tags around each individual question, allowing a learner to mark each one in turn. **Shuffling questions**: if the shuffle questions button is shown, the questions can be shuffled by clicking the button. This only works so long as the quiz-taker has not started to answer the questions. Once the quiz-taker starts answering, the order is locked.Shuffle button is not shown if there is one or less questions in a quiz.
<quiz display=simple> *insert questions here* </quiz>	*This is where questions would appear.* Submit	• Importantly, the addition of "display=simple" as an attribute of the quiz tag removes the points table at the top. • This is useful for short quizzes, and especially for situations where you insert single questions between text blocks.
<quiz shuffle=none> *insert questions here* </quiz>	Point added for a correct answer: 1 Points for an incorrect answer: 0 Ignore the questions' coefficients: ☐ *This is where questions would appear.* Submit	• If you'd like to keep the points table but get rid of the button for shuffling questions, use *shuffle=none* as shown.

<quiz shuffleanswers=true>
questions with proposal shuffled on reload here
</quiz>

Point added for a correct answer: 1

Points for an incorrect answer: 0

Ignore the questions' coefficients: ☐

Shuffle questions

1 Sample Question 1 ⦿
○ Distractor.
○ Distractor.
○ Distractor.
○ The correct answer.

2 Sample Question 2 ⦿
○ Distractor.
○ Distractor.
○ The correct answer.
○ Distractor.

Submit

Shuffling answers: The quiz also provides the option of automatic shuffling of answers on load.This can be achieved by using shuffleanswers parameter.When shuffleanswers parameter is set to true the answers are shuffled for each reload of the page. This only works so long as the quiz-taker has not started to answer the questions. Once the quiz-taker starts answering, the order is locked to the way it was before submission.

Inserting questions

The types of questions available are shown below.

An important point is that **each question must be separated by a blank line**, otherwise they are merged together into one question.

How to use the examples

In the examples on this page, there is both *wikitext* and *rendering*. The *wikitext* shows you the code, which you can copy and paste into your pages, or just learn and understand. The *rendering* shows you what the result will look like.

Copying and pasting

The markup has been slightly simplified in order to help *understanding*. If you copy and paste, you must insert <quiz> tags before and after the markup. You may wish to use the quiz tags with the "display=simple" option as shown above.

Note that the actual example below also use *tables* for clarity. If a quiz is inserted into a table, this modifies the appearance of a quiz - for example, boxes appear around some elements in the quiz. If you don't use a table, these boxes disappear and your quiz will look plainer.

When pasting, ensure that **all the symbols are aligned with the beginning of the lines of the page** - i.e. if any line begins with a space, formatting errors may occur.

Live demos

All the examples below should work. Don't be afraid to try them out. Hitting the *correction* button is OK - it won't change or damage this page.

Basic question types

- **Note**: For multiple choice questions(MCQs) with negative(-) sign and positive(+) sign, the positive sign(+) signifies the correct answer while the negative(-) sign(s) signifies the wrong answer and this is how Quiz know which answer is correct and which one is wrong.

Multiple choice / single response

wikitext	rendering	comments

`{Question |type="()"}`
`+ The correct answer.`
`- Distractor.`
`- Distractor.`
`- Distractor.`
	Question
	○ The correct answer.
	○ Distractor.
	○ Distractor.
	○ Distractor.
	Submit
		• This type of question allows a **single** correct response and any number of distractors (3 distractors are shown, but it could be one, or nine, or any other number).
		• The correct answer may be in any position, and is marked by a plus sign. Distractors are marked by minus signs.
		• The wikitext shown should be inserted between <quiz> tags as shown above.
		• Answers appear in the same order as in the markup, so it is advisable to shuffle them before exporting into markup.
		• Every symbol, i.e. |, +, - and any other symbols listed further down, **must** be on a new line.
		• The symbols must **begin** the newline - no spaces before them.
		• The borders are a result of the border around the wiki table. They are optional.
		• A *distractor* is the standard term for a multiple choice option which is wrong.
		• The () symbol in the type attribute is supposed to look like a radio button.
		• To see the full colour code (with four colours), try doing the quiz and clicking the correction button.
		• A *second* click on a radio button (i.e. a click on an already clicked radio button) will deselect it.

True/false questions

wikitext	rendering	comments

wikitext	rendering	comments
{Bulgaria and Romania joined the European Union in 2007. |type="()"} + TRUE. - FALSE. {Switzerland is a member of the European Union. |type="()"} - TRUE. + FALSE. {Anyone can be a Citizen Journalist. |type="()"} + TRUE. - FALSE.	1 Bulgaria and Rumania joined the European Union in 2007. ○ TRUE. ○ FALSE. 2 Switzerland is a member of the European Union. ○ TRUE. ○ FALSE. 3 Anyone can be a Citizen Journalist. ○ TRUE. ○ FALSE. Submit	• In many quiz systems, true/false questions are listed as a separate kind of question, although they are really nothing more than a variety of multiple choice with a single response. The Wikiversity quiz extension correctly treats true/false questions as multiple choice with a single response. They are implemented in exactly the same way as above. • The first of the two examples shows a true/false question where the correct answer is "true". • The second of the two examples shows a true/false question where the correct answer is "false". • The examples are live and can be tried out now. • If you have a large number of true/false questions, a more efficient way to lay them out would be to use the tabular response type with radio buttons documented below.

Multiple choice / multiple response

wikitext	rendering	comments
{Question |type="[]"} + Correct answer. - Incorrect answer. + Correct answer. - Incorrect answer.	Question ☐ Correct answer. ☐ Incorrect answer. ☐ Correct answer. ☐ Incorrect answer. Submit	• This type of question is a multiple choice where more than one answer could be correct. • Correct answers are indicated by plus signs, incorrect ones by minus signs. • The wikitext shown should be inserted between <quiz> tags as shown above. • Answers appear in the same order as in the markup, so it is advisable to shuffle them before exporting into markup. • Every symbol, i.e. |, +, - and any other symbols listed further down, **must** be on a new line. • The symbols must **begin** the newline - no spaces before them. • The borders are a result of the border around the wiki table. They are optional. • The [] symbol in the type attribute is supposed to look like a checkbox.

Typed response (short answer)

wikitext	rendering	comments

```
{  
|type="{}"}
Where was Aristotle
born?
{ Stageira }

{  
|type="{}"}
Name Aristotle's
famous mentor.
{ Plato }

{  
|type="{}"}
{ Alexander } was
Aristotle's most
famous pupil.
```

1

Where was Aristotle born?

2

Name Aristotle's famous mentor.

3

was Aristotle's most famous pupil.

Submit

- In a *typed response* the quiz-taker types an answer into a textfield using the keyboard.
- The question type is signalled using **"{}"** as the argument of the *type*.
- A textfield is inserted using curly brackets at the desired position of the textfield.
  - Note that curly brackets alone don't create a textfield. The curly brackets must have spaces between them and the enclosed text. **{ Answer }** works, but **{Answer}** doesn't work.
  - The correct answer must be inserted between the curly brackets.
  - The textfield can be positioned anywhere in the text. Text before and after the text field will display as normal text.
- The borders are a result of the border around the wiki table. They are optional.
- A non-breaking space was inserted between { and |type= because some text is required at that position (slightly buggy thing).

```
{
|type="{}"}
Name a famous Greek
philosopher.
{ Aristotle|Plato }
```

Name a famous Greek philosopher.

Submit

- Typed responses often have more than one possible answer. You can list all the possible answers inside the curly brackets using a pipe to separate them.
- In the example, there are two possible answers. Try it - the demo is live.

```
{
|type="{}"}
Name Aristotle's
famous mentor.
{ Plato _6 }
```

Name Aristotle's famous mentor.

Submit

- To modify the size of the textfield, add an underscore and an integer following the answer within the curly brackets to represent the number of glyph spaces you need.
- This number depends, however it should be around n+2 (where n is the number of glyphs in the answer)

```
{ Complete each box
with a synonym
of the following
words
|type="{}"}
#free: { gratis _6 }
#well: { good _4 }
```

Complete each box with a synonym of the following words

1. free:
2. well:

Submit

- You can group the items into one only question.

```
{
|type="{}"}
Where did Aristotle
die?
{ Chalcis (i) }
```

Where did Aristotle die?

Submit

- **Case-sensitivity**: typed responses are *case-sensitive* unless you add ***(i)*** at the position shown. The (i) stands for case-**in**sensitive.
- If some question have more than one possible answer, you should include one ***(i)*** for each of them.

Gapfill / cloze

wikitext

```
{Complete the text from the Wikipedia article on
[[w:Aristotle|Aristotle]].
|type="{}"}
Aristotle was born in { Stageira _8 } in Chalcidice. His parents
were { Phaestis _8 }
and { Nicomachus _10 }, who became physician to King Amyntas of
Macedon. Aristotle
was educated as a member of the aristocracy. At about the age of {
eighteen|18 _8},
he went to { Athens _6 } to continue his education at { Plato _5
}'s Academy.
Aristotle remained at the Academy for nearly twenty years, not
leaving until after
Plato's death in { 347 _3 } BC.
```

rendering

Complete the text from the Wikipedia article on Aristotle.
Aristotle was born in
in Chalcidice. His parents were
and
, who became physician to King Amyntas of Macedon. Aristotle was educated as a member of the aristocracy. At about the age of
, he went to
to continue his education at
's Academy. Aristotle remained at the Academy for nearly twenty years, not leaving until after Plato's death in
BC.
Submit

comments

• Gapfills (or cloze exercises) are created in almost exactly the same way as typed responses. The difference is that you just write a bit more text and add a few more textfields.
• The gapfill above was additionally placed in a box and centered using a wiki table.

Numerical response

wikitext	rendering	comments

`{In which year was Plato born? |type="{}"} He was born in { 427-428 _5 } BC.`	In which year was Plato born? He was born in BC. Submit	• At Wikiversity, numerical responses are a subtle modification of typed responses. The syntax is identical, except for some additions which allow numerical ranges and numerical tolerances if the answer is a number. • In the example to the right, the uncertainty over Plato's date of birth is reflected in the answer by using a **range**. A simple hyphen between the numbers permits all numbers inbetween.
`{How far is the earth from the sun? |type="{}"} { 91-95 _3 } million miles.`	How far is the earth from the sun? million miles. Submit	• This is another example of a range. • The _3 ensures that the textfield doesn't get too wide.
`{What is 10*10? |type="{}"} { 100 5% _3 }`	What is 10*10? Submit	• This is an example of specifying a **tolerance**. The critical part is "5%", which specifies the expected tolerance of 5%. In this example, values between 95 and 105 should work, but try it. • At the time of writing, the underscore method of limiting the size of the textfield does not yet work in combination with a tolerance specification, so a range may be a better method of specifying numerical accuracy.
Complete the following text, inserting words and numbers as appropriate. The Earth formed around billion years ago and its only known natural satellite, the , began orbiting it around billion years ago. At present, the Earth orbits the once for every roughly times it rotates about its axis. Submit		• Because numerical responses are, programmatically, a sub-variety of typed response, numerical and text responses can be freely mixed in a gapfill as shown to the left.

Tabular response type with radio buttons

wikitext	rendering	comments

wikitext	rendering	comments

`{Question`
`|type="()"}`
`| A | B | C`
`+-- A is correct.`
`-+- B is correct.`
`--+ C is correct.`
	Question
	A B C
	○ ○ ○ A is correct.
	○ ○ ○ B is correct.
	○ ○ ○ C is correct.
	Submit
		• Add extra radio buttons by reduplicating the +/- signs at the beginning of each row.
		• There should only be one plus sign in each row.
		• Specify columns headers by using pipes at the top.

Tabular response type with checkboxes

wikitext	rendering	comments

`{Question`
`|type="[]"}`
`| Option A | Option B | Option C`
`++- Question 1 (Clue: A and B are correct).`
`-++ Question 2 (`
`Clue: B and C are correct).`
`--+ Question 3 (Clue: C is correct).`
	Question
	Option A **Option B** **Option C**
	☐ ☐ ☐ Question 1 (Clue: A and B are correct).
	☐ ☐ ☐ Question 2 (Clue: B and C are correct).
	☐ ☐ ☐ Question 3 (Clue: C is correct).
	Submit
		• This type of question has multiple everything.
		• Correct answers are indicated by plus signs, incorrect ones by minus signs. By reduplicating the +/- signs at the beginning of each row, extra checkboxes can be added.
		• The column headers are defined using pipes at the top as shown.

Advanced stuff

Inserting feedback

wikitext	rendering	comments

`{Question`
`|type="()"}`
`+ The correct answer.`
`|| Feedback for correct answer.`
`- Distractor.`
`|| Feedback for distractor.`
`- Distractor.`
`|| Feedback for distractor.`
`- Distractor.`
`|| Feedback for distractor.`
	Question
	○ The correct answer.
	○ Distractor.
	○ Distractor.
	○ Distractor.
	Submit
		• Feedback is text which is initially hidden, and then shown to the user when they submit the quiz for correction. Feedback typically serves the purposes of encouragement, explanation, congratulation and commiseration.
		• Feedback is set with all types by using a double vertical line **||** immediately after a response.
		• The pipes beginning the *feedback* section must be on a new line.
		• Unlike some other quiz systems, **all** feedback items are displayed on correction. With other quiz systems, the norm is to display the feedback only for those items which were selected by the user.
		• To try out the feedback effect, choose a response and click the correction button. The feedback will then appear.

Inserting blocks of text between questions

wikitext	rendering	comments

```
{This is the
first question.
|type="()"}
+ The correct
answer.
- Distractor.
- Distractor.

{This is a block
of text.}

{This is the
second question.
|type="()"}
+ The correct
answer.
- Distractor.
- Distractor.
```

1 This is the first question.

○ The correct answer.

○ Distractor.

○ Distractor.

Arctic Terns

The **Arctic Tern** is a seabird of the tern family Sternidae. This bird has a circumpolar distribution, breeding colonially in Arctic and sub-Arctic regions of Europe, Asia, and North America (as far south as Brittany and Massachussets).

2 This is the second question.

○ The correct answer.

○ Distractor.

○ Distractor.

Submit

• Text blocks can be inserted between questions by placing it between curly brackets {....}.
• The text blocks can contain wiki markup, such images, hyperlinks and formatting tags, as shown in this example.

Using images as answers

wikitext	rendered	comments

`<quiz display=simple>`
`{ Which of these are not nuts? }`
`- [[Image:Walnut03.jpg|100px]]`
`+ [[Image:Peanuts.jpg|100px]]`
`+ [[Image:Malus-Idared on tree.jpg|100px]]`
`- [[Image:Pistachios in a bag.jpg|100px]]`
`- [[Image:Chestnut.jpg|100px]]`
`+ [[Image:P10802511.JPG|100px]]`
`</quiz>`
	Which of these are not nuts?
	☐ 100px
	☐ 100px
	☐ 100px
	☐ 100px
	☐ 100px
	☐ 100px
	Submit
		• Unfortunately there does not yet seem to be a way of putting these into a row or tabularising the layout.
		• Although this has not been done here for reasons of space, it is generally good practice to specify some kind of tooltip text for the images, but try to avoid text which gives away the answer.

Using images in the question

wikitext	rendered	comments

`<quiz display=simple>`
`{ What is this?`
`[[Image:Butterfly tongue.jpg|100px| ]]`
`| type="()" }`
`- Tiger`
`+ Butterfly`
`- Bee`
`- Fly`
`</quiz>`
	What is this? File:Butterfly tongue.jpg
	○ Tiger
	○ Butterfly
	○ Bee
	○ Fly
	Submit
		In this example, the image is part of the question. You just use normal wiki-markup to insert the image(s).

Using animations

The section adds no new technical tricks, but it looks pretty.

File:Heringsschwarm.gif
What kind of fish are these?

- ○ Herring
- ○ Salmon
- ○ Tunafish
- ○ Goldfish

Submit

File:Radial engine.gif
What's this?

- ○ a boxer engine
- ○ a wankel engine
- ○ a radial engine

Submit

File:Pyramid of 35 spheres animation.gif
How many balls are there here?

Submit

Hyperlinks and other markup

Markup that works inside quizzes

- Hyperlinks: [[....]].
- Images: [[Image:....]].
- Other media: [[....]].
- Bold and italic.
- <big> and

Markup that partly works inside quizzes

- Templates: {{....}}.
  - *subst* doesn't work.
- Tables: {|....|}.
- Headings: e.g. ===....===.
  - **Warning:** At the moment there is a problem with the toc.
- Lists and indents: using #, * and :

Adding music and sound effects

This is similar to insertion of images. Standard wiki markup is used.

Configuring points

wikitext (what you write)	rendering (what you see)	comments

`<quiz points="3/1!">`
`{Try this question now.}`
`+ Plus 3 points for this one.`
`- Minus 1 point for this one.`
`</quiz>`
	Points added for a correct answer: 3
	Point for an incorrect answer: 1
	Ignore the questions' coefficients: ☒
	Try this question now.
	☐ Plus 3 points for this one.
	☐ Minus 1 point for this one.
	Submit
		• By default, there is 1 point for a correct answer and 0 for a wrong answer.
		• You can change this for the whole quiz by using a *points* attribute for the *quiz* tag.
		• The format is "X/Y!", where X gives the points for a correct answer and Y the points for a wrong answer. The Y points are expressed as a positive integer but are *subtracted* from the score.
		• Values like 1.25 or 2.5 are allowed

```
<quiz
display=simple>
{Try this question
now.
|type="()" }
+ 1 points for
this one.
- 0 points for
this one.

{Try this question
now.
|type="()"
coef="2"}
+ 2 points for
this one.
- 0 points for
this one.

{Try this question
now.
|type="()"
coef="4"}
+ 4 points for
this one.
- 0 points for
this one.

{Try this question
now.
|type="()"
coef="0.5"}
+ 1/2 point for
this one.
- 0 points for
this one.
</quiz>
```

1 Try this question now.

○ 1 point for this one.

○ 0 points for this one.

2 Try this question now.

○ 2 points for this one.

○ 0 points for this one.

3 Try this question now.

○ 4 points for this one.

○ 0 points for this one.

4 Try this question now.

○ 1/2 point for this one.

○ 0 points for this one.

Submit

• A *coefficient* is a way of giving a weighting to a specific question. Normally all the questions have the same number of points. By default, they all have one point. By specifying a coefficient for each or any question, you can multiply the number of points for that question by the coefficient.
• Coefficients can be both numbers above one and numbers between 0 and 1. If the coefficient is above 1, the question will receive greater weighting. If the coefficient is less than 1, the question will receive less weighting.
• You **must** put inverted commas around the argument of the *coef* attribute.

Shuffle Control Tags

wikitext	rendered	comments

```
<quiz>
{X}

{ Sample Question 1
|type="()"}
+ The correct answer.
- Distractor.
- Distractor.
- Distractor.
{ Sample Question 2
|type="()"}
+ The correct answer.
- Distractor.
- Distractor.
- Distractor.

{\X}
{!X}

{ Sample Question 3
|type="()"}
+ The correct answer.
- Distractor.
- Distractor.
- Distractor.
{ Sample Question 4
|type="()"}
+ The correct answer.
- Distractor.
- Distractor.
- Distractor.

{\X}
</quiz>
```

Point added for a correct answer: 1

Points for an incorrect answer: 0

Ignore the questions' coefficients: ☐

Shuffle questions

1 Sample Question 1

- ○ The correct answer.
- ○ Distractor.
- ○ Distractor.
- ○ Distractor.

2 Sample Question 2

- ○ The correct answer.
- ○ Distractor.
- ○ Distractor.
- ○ Distractor.

3 Sample Question 3

- ○ The correct answer.
- ○ Distractor.
- ○ Distractor.
- ○ Distractor.

4 Sample Question 4

- ○ The correct answer.
- ○ Distractor.
- ○ Distractor.
- ○ Distractor.

Submit

Conditional shuffling using tags:It can be achieved by using {X},{!X} and {/X} tags inside quiz.Any questions that are within {X} and {/X} are treated as a block and will be shuffled.The questions that are within {!X} and {/X} are not shuffled internally, i.e the order of questions inside {!X} and {/X} remain the same.
In the example Sample Question 1 and 2 can be shuffled but Sample Question 3 and 4 are always in the same order.

Troubleshooting

- You **cannot** test out the quiz while editing a page, because the *submit* button is grayed out. If you wish to test your new quiz, then create a sandbox page somewhere else first (e.g. as a subpage of your user page), then test the quiz there, and finally transfer the quiz to its final location (and test it again anyway).
- The quiz was developed for IE6, IE7, Firefox (and Opera?), but display quality on Opera may not be optimal, and there may be display differences between browsers.
- A reminder: each question must be separated by a blank line, otherwise the questions are merged together into one question.

Future stuff

Information about future developments is mainly available at the project space on MediaWiki. Lots of things are in planning.

Back to top of page - Back to Welcome Page

Mediawiki Extension RenameUser

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The Renameuser extension provides a special page which allows authorized users to rename user accounts. This will cause page histories, etc. to be updated. If you want to rename an account "into" another, already existing one, you need the User Merge and Delete extension.

Installation

Download

Download and place the file(s) in a directory called Renameuser in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'Renameuser' );
```

Configuration

It is good practice to allow sysop to use it and not only the embers of the Burocrat group.

```
$wgGroupPermissions['sysop']['renameuser'] = true;
```

Usage

As a bureaucrat, access the Special:Renameuser page.
Enter the name of the user to be renamed, and the desired new name, toggle the options to move user/talk pages and their subpages and to create redirects as desired, and click OK.
The user will be renamed, and (if the appropriate option was checked) the user page, user talk page and all their subpages will be moved as well.

The ability of this extension to move user subpages does not depend on 'move-subpages' userright. The number of subpages moved is not limited by the $wgMaximumMovedPages parameter. The extension will also move all css, js and move-protected pages.
You can grant other groups the right to rename users, for example:

```
$wgGroupPermissions['sysop']['renameuser'] = true;
```

See also

User Merge and Delete extension

Back to top of page - Back to Welcome Page

Mediawiki Extension Restrict Access By Category and Group

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

This extension can restrict access to users by group and document category.

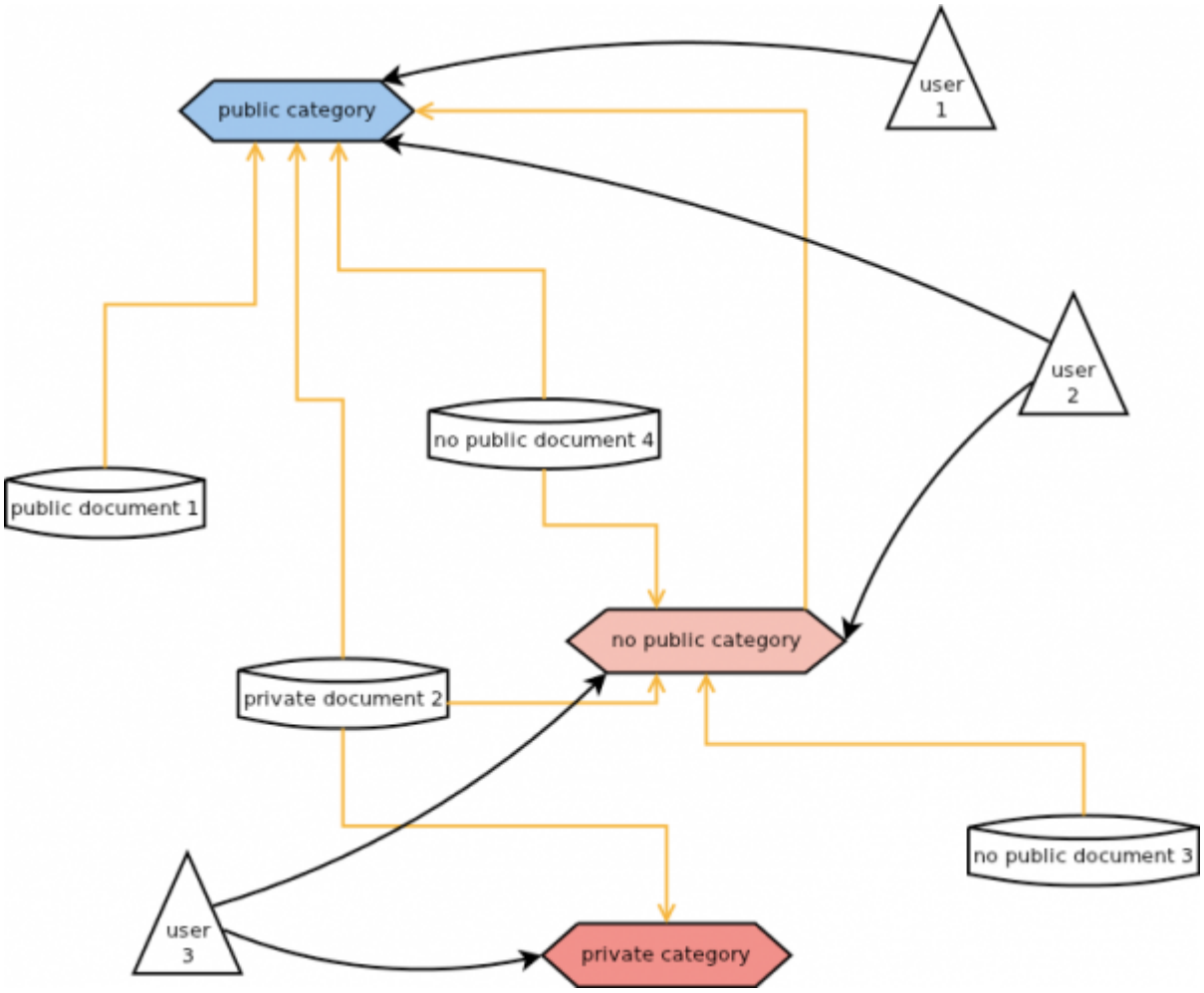
Even though Mediawiki is a free/public access collaborative document tool, sometimes it can be helpful, especially in business environments, to have a restricted view of wiki documents. Especially for a private wiki.

For example, department's financial documents should not be accessed by customer services users.

In this extension you can establish four access restrictions:

- Public: White pages: those are public pages that can be accessed by everybody. This is helpful when you have a private mediawiki and anonymous can only authenticate and see Main page. See $wgWhitelistRead.
- Public categories: those are all categories that aren't in your groups.php file. Those categories are public.

- No public categories: those are all categories that are in your groups.php file. Those categories are restricted. Documents that belong to these categories may be accessed by users who belong to at least one of these groups.
- Private categories:those are all categories that are in your groups.php file with [private] = true option. These categories are private, and only users who belong to one or more of this categories closed to which the document will have access. It is the same behavior that if the document did not belong to any of the no public categories basis only membership to the private category (respect to access).



- user1 could access to public document 1 only.
- user 2 could access to public document 1, no public document 3 and no public document 4.
- user 3 could access to public document 1, no public document 3, no public document 4 and private document 2.

Usage

All documents you want to restrict access to, you must add to a category. So you have multiple categories.

Files:

- $IP/extensions/rabcg/rabcg.php: this is the extension.
- $IP/extensions/rabcg/groups.php: this is the group catalog.

In your groups.php file, you must add the categories you want to make no public or private. This is made by group definition.
For Example:

```
<?php
// This is a no public category: Financial no public data.
$wgGroupPermissions['Financial no public data']['*'] = true;
// This is a private category: Financial private data.
$wgGroupPermissions['Financial private data']['private'] = true;
```

To apply this category to your document, you only must write:

This is only one more category of your document.

Previously (after create the groups in you groups.php file), you must make groups assignments to users by Special:UserRights page.

Download instructions

This extension is not yet in MediaWiki SVN Repository.

Therefore, you must copy & paste the following code as is explained in the installation section below.

Installation

Folder creation

Create a folder in your root/extensions folder :

```
sudo mkdir RestrictAccessByCategoryAndGroup
```

Create the extension php file

```
sudo nano
RestrictAccessByCategoryAndGroup/RestrictAccessByCategoryAndGroup.php
```

And insert the following code:

```
<?php
if ( !defined( 'MEDIAWIKI' ) ) {
        die( 'Not a valid entry point.' );
}
$wgExtensionCredits['parserhook'][] = array(
        'name' => 'Restrict access by category and group',
        'author' => 'Andrés Orencio Ramirez Perez',
        'url' =>
'https://www.mediawiki.org/wiki/Extension:Restrict_access_by_category_and_gro
```

```
up',
        'description' => 'Allows to restrict access to pages by users groups
and page categories',
        'version' => '2.0.1'
);
$wgHooks['userCan'][] = 'restrictAccessByCategoryAndGroup';
function restrictAccessByCategoryAndGroup( $title, $user, $action, $result )
{
        global $wgGroupPermissions;
        global $wgWhitelistRead;
        global $wgLang;
        global $wgHooks;
        global $wgContLang;
        global $wgWhitelistRead;
        global $wgVersion;
//The Main Page, Login and Logout pages should always be accessible
        if ( $wgVersion >= '1.17' ) {
                $wgWhitelistRead[] = wfMessage( 'mainpage' )->plain();
        } else {
                $wgWhitelistRead[] = wfMsgForContent( 'mainpage' );
        }
        $wgWhitelistRead[] = SpecialPage::getTitleFor( 'Userlogin'
)->getLocalUrl();
        $wgWhitelistRead[] = SpecialPage::getTitleFor( 'Userlogout'
)->getLocalUrl();
$validCategory = false;
        $groupExists = false;
        $pageHasCategories = false;
        $privateCategory = false;
        $privateCategoryTemp = false;
        $categoryNamespace = $wgLang->getNsText( NS_CATEGORY );
        $whitePage = true;
//System categories
        $systemCategory = array();
        foreach ( array_change_key_case( $title->getParentCategories(),
CASE_LOWER ) as $key => $value ) {
                $formatedKey = substr( $key, ( strpos( $key, ":" ) + 1 ) );
                $systemCategory[ $formatedKey ] = $value;
        }
//Is this page a white page?
        if ( isset( $wgWhitelistRead[0] ) ) {
                $whitePage = in_array( $title, $wgWhitelistRead );
        }
//If the page has no categories, it's public.
        if ( count( $title->getParentCategories() ) == 0 ) {
                $validCategory = true;
        } else {
                //For each system categories
                foreach ( $wgGroupPermissions as $key => $value ) {
                        //If current system category is defined as private,
then tmpCatP is true
```

```
                        if ( isset( $wgGroupPermissions[ $key ]['private'] )
) {
                                $privateCategoryTemp = $wgGroupPermissions[
$key ]['private'];
                        } else {
                                $privateCategoryTemp = false;
                        }
                        //If current system category exist in the document
category array ...
                        if ( array_key_exists( strtolower( str_replace( " ",
"_", $key ) ), $systemCategory ) ) {
                                if ( $privateCategoryTemp
and !$privateCategory ) {
                                        $privateCategory = true;
                                        $validCategory = false;
                                }
                                //We see that the user belongs to one of the
groups (like of category)
                                if ( in_array( $key, $user->getGroups() ) and
( !$privateCategory or ( $privateCategoryTemp and $privateCategory ) ) ) {
                                        $validCategory = true;
                                }
                                $groupExists = true;
                        }
                }
                $pageHasCategories = count( $title->getParentCategories() ) >
0;
        }
if ( !$pageHasCategories ) {
                return true;
        }
        if ( !$groupExists and !$whitePage ) {
                return true;
        }
        if ( ( $user->isLoggedIn() and $validCategory ) or $whitePage ) {
                return true;
        }
        return false;
}
```

Edit Localsettings.php

Activate the extension

Add the following to your LocalSettings.php

```
require_once
"$IP/extensions/RestrictAccessByCategoryAndGroup/RestrictAccessByCategoryAndG
roup.php";
```

Add the security groups

Create the groups as needed like the example here below. Make sure you keep track of your security and access matrix, this can very rapidly become extremely complex.

```
$wgGroupPermissions['Financial no public data']['*'] = true;
$wgGroupPermissions['Financial private data']['private'] = true;
```

Back to top of page - Back to Welcome Page

Mediawiki Extension RightFunctions

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The RightFunctions extension tests various page or user permissions and displays text based off of them.

Please note that these parser functions may be used to superficially hide information from those with certain rights. Also note that all functions disable the cache on the page they are used in unless you set $wgRightFunctionsAllowCaching to true.

Installation

Download

Download and place the file(s) in a directory called RightFunctions in your extensions/ folder.

Version 1.35 LTS

Activation

Add the following code at the bottom of your LocalSettings.php:

```
require_once "$IP/extensions/RightFunctions/RightFunctions.php";
```

Configuration

These variables control how the various parser functions work. There are currently 4 variables.

- $wgRightFunctionsUserGroups : Default value: array('*', 'user', 'autoconfirmed', 'sysop', 'bureaucrat').
  An array of groups to test for in #Usergroup. They are arranged in order of least to greatest for group testing.
- $wgRightFunctionsAllowCaching : Default value: false.
  If set to true, the functions do not disable the parser cache and they do not trigger the recache when getting a user's groups. It is recommended that you leave this as false.
- $wgRightFunctionsAllowExpensiveQueries : Default value: true.
  If set to true, using ifpageright and ifpageallowed will check the rights more thoroughly (like through cascading protection), so it will produce more accurate results, but it will me more server-intensive to perform. It is recommended that you leave this as true.
- $wgRightFunctionsDisableFunctions : Default value: array();
  Functions in this array will not get evaluated and will simply return nothing.

Usage

Ifright

Tests whether the current user has the given right, then displays the appropriate text. If the right doesn't exist, it simply returns else. If no right is specified, it returns then. This function ignores per-page restrictions on a user's rights, so use #Ifpageright if you need that functionality.
Usage: {{#ifright:right|then|else}}

Ifallowed

Tests whether the given user has the given right, then displays the appropriate text. This function will return nothing unless a user is defined. This function ignores per-page restrictions on a user's rights, so use #Ifpageallowed if you need that functionality.
Usage: {{#ifallowed:user|right|then|else}}

Switchright

A switch statement that tests whether the current user has the rights. This operates much like the ParserFunctions #switch statement.

Usage: {{#switchright:right1=result1|right2|right3=result2 and 3|right4=result4|default (or #default=default if the default text contains =)}}

Userrights

Lists all rights in an unordered list either of the current user or of the given user. If a user isn't defined, it triggers on the current user instead.

Usage: {{#userrights:}} (for current user) or {{#userrights:user}} (for other users)<//nowiki>

=== Usergroup === Returns the highest user group the user belongs to. The groups are defined in lowest-to-highest ranking in the array $wgRightFunctionsUserGroups. If a user isn't defined, it triggers on the current user instead.
 Usage: <nowiki>{{#usergroup:}} (for current user) or {{#usergroup:user}} (for other users)

Ifgroup

Tests if the given user is in the specified group. If no user is specified, it tests the current user. If no group is specified, it returns else.

Usage: {{#ifgroup:group|then|else}} (for current user) or {{#ifgroup:group|then|else|user}} (for other users)

Switchgroup

A switch statement that tests whether the current user is in a group. This operates much like the ParserFunctions #switch statement.

Usage: {{#switchgroup:group1=result1|group2|group3=result2 and 3|group4=result4|default (or #default=default if the default text contains =)}}

Ifpageright

Tests whether the current user has the given right on the given page, then displays the appropriate text. If the right doesn't exist, it simply returns else. If no right is specified, it returns then. This function does not ignore per-page restrictions on a user's rights.

Usage: {{#ifpageright:right|then|else|page}} for testing other pages or {{#ifpageright:right|then|else}} for testing the current page.

Ifpageallowed

Tests whether the given user has the given right on the given page, then displays the appropriate text. If no user is specified, it returns nothing. If no page is specified, it checks the current page. If the right doesn't exist, it returns else and if no right is specified, it returns then. This function does not ignore per-page restrictions on a user's rights.

Usage: {{#ifpageallowed:user|right|then|else|page}} for testing other pages or {{#ifpageallowed:user|right|then|else}} for testing the current page.

Ifprotected

Tests whether the given page is protected, then displays the appropriate text. If no page is given, it tests the current page. It will return then if the page is either fully-protected, semi-protected, protected because it is included in a cascading-protected page, or if it is namespace-protected. You may control which of these are tested by adding the letters "f", "s", "c", and/or "n" (standing for fully-protected, semi-protected, cascading-protected, and namespace-protected, respectively) to the fourth parameter of the page (by default all 4 are tested).

Usage: {{#ifprotected:then|else|page}} for testing other pages, or {{#ifprotected:then|else}} for testing the current page or (for example) {{#ifprotected:then|else|page|fns}} for testing only certain types of protection.

Getrestrictions

Grabs what right is required to carry out the action on the given page. If no page is given, it checks the current page instead. If no right is given, it checks the 'edit' right. If there are no restrictions on that page for the particular right, it returns the empty string (""). If the page has cascading restrictions as well, it returns the highest restriction level for the page (from all cascading sources and the local page itself). If you would like all the restrictions to be listed instead, set the third parameter to "true". If you do, the output format changes to be more detailed, depending on what restrictions exist for the current page, cascading-protected sources, and namespace protection.

Usage: {{#getrestrictions:right|page}} for testing other pages, or {{#getrestrictions:right}} for testing the current page or {{#getrestrictions:right|page|true}} for listing all restrictions instead of just the highest one.

Back to top of page - Back to Welcome Page

Mediawiki Extension Semantic Mediawiki

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Pre-requisite

Composer needs to be installed and available first. Information about Composer installation is available here.

Installation

Since you should already have composer installed, you can do this by running composer in the top-level MediaWiki directory. Assuming you have composer.phar in that directory, the following command will work:

```
COMPOSER=composer.local.json php composer.phar require --no-update
mediawiki/semantic-media-wiki
```

Alternatively, if the above fails, you can add the following content to your composer.local.json file:

```
{
```

```
        "require": {
                  "mediawiki/semantic-media-wiki": "~4.0"
        }
}
```

Now run the following command from the base directory1 of your MediaWiki installation:

composer update --no-dev

Enabling Semantic Mediawiki

Add a call to wfLoadExtension() as well as enableSemantics() to the end of the "LocalSettings.php" file.
enableSemantics() takes in the domain name of the wiki; a wiki located at "example.org", for instance, should have the following call:

```
wfLoadExtension( 'SemanticMediaWiki' );
enableSemantics( 'example.org' );
```

Setup and update

Run MediaWiki'S "update.php" script from the base directory1 of your MediaWiki installation:

```
php maintenance/update.php
```

Verify

Check the "Special:Version" page on your wiki and verify that the extension has been successfully installed and enabled.

Testing

Create a regular wiki page named "TestSMW".
Before saving, enter the following wiki text within the edit field:

```
        Property test: [[Property name::Property value]]
```

Now save the page and you will see the following displayed on the page:

"Property test: Property value"

Navigate to special page "Special:Browse" by clicking on the link "Browse properties" in the "tools" section of your wiki's sidebar.
On this special page you will see:

> the property "Propery name" and the value "Propery value", both in red
> the special property "Modification date" in blue and italics as well as the value e.g. "5 February 2014 23:42:29" (the timestamp when the page was saved).

Semantic MediaWiki was successfully tested!

Mediawiki Extension ShowMe

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The ShowMe extension allows inserting a dropdown into pages, that will show different elements on the page depending on which option is selected.

Installation

Download

Download and place the file(s) in a directory called ShowMe in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'ShowMe' );
```

Usage

Use the <showme> tag. It has an optional name parameter, which can set the name and id of the <select> element. Each line between the tags should contain an option. It must consist of the text to be shown, followed by a pipe, followed by the class name of an element (or elements) on the page which should be shown when this option is selected.
While one option is selected, the elements corresponding to all other options become hidden. To avoid flashing of elements that are supposed to be hidden, you can optionally add style="display: none;" to every option other than the first.

Example

Dropdown

```
<showme type="dropdown">
Text for Option 1|opt-1
And another option|opt-2
</showme>
<div class="opt-1">
```

```
Some text shown when clicking the first option.
</div>
<div class="opt-2" style="display: none;">
Here goes the text shown for the second option.
</div>
```

Unordered list

```
<showme type="ul">
Text for Option 1|opt-1
And another option|opt-2
</showme>
<div class="opt-1">
Some text shown when clicking the first option.
</div>
<div class="opt-2" style="display: none;">
Here goes the text shown for the second option.
</div>
```

Back to top of page - Back to Welcome Page

Mediawiki Extension UserMerge

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The UserMerge extension allows wiki users with the usermerge permission (Bureaucrat by default) to merge one Wiki user's account with another Wiki user's account.

The extension allows to :

- merge (refer contributions, texts, watchlists, edit count of a first account A to a second account B)
- delete the first account A after the merge (option box)

Limitations

- you cannot delete a user A without having merged the user A to B
- you cannot merge your own account (logged-in user) into a different user

Extras

- if you omit the "New User" field, then the extension auto-populates the New User as "Anonymous" (user_id 0), and ask you to confirm a merge to Anonymous. This is used for user deletion: you first empty (merge to user 0) contributions of a user A, and then delete the user A.
    - If your version of the extension complains about an empty "New User" field you can fill in "Anonymous"
- the extension creates a User Merge log and logs all User Merge extension activity.[1]

Installation

Download and place the file(s) in a directory called UserMerge in your extensions/ folder.

Download

Download and place the file(s) in a directory called UserMerge in your extensions/ folder.

Activation

Add the following code at the bottom of your LocalSettings.php:

```
wfLoadExtension( 'UserMerge' );
```

Configuration

```
// By default nobody can use this function, enable for bureaucrat?
$wgGroupPermissions['bureaucrat']['usermerge'] = true;
// optional: default is array( 'sysop' )
$wgUserMergeProtectedGroups = array( 'groupname' );
```

Back to top of page - Back to Welcome Page

Mediawiki Extension WhosOnline

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Mediawiki Extension
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Description

The WhosOnline extension adds a special page - Special:WhosOnline - for viewing the list of users currently viewing a page on the wiki.

Installation

Download

Download the extension from here.

Activation

Set this in your LocalSettings.php file

```
wfLoadExtension( 'WhosOnline' );
```

Configuration

Set

```
$wgWhosOnlineShowAnons = true;
```

If you want anonymous users IP addresses shown. Default is not to show them, for privacy reasons.

Usage

Users who login after the extension is installed can be seen at Special:WhosOnline. Refresh the page if it is blank. As of version 1.3.1 it is possible to include the contents of the special page into a page: {{:Special:WhosOnline}}. If you use {{:Special:WhosOnline/10/shownav}} it will only show 10 users and will have the navigation controls.

Back to top of page - Back to Welcome Page

Mediawiki Security Matrix

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Security
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Micylou Standard Mediawiki Security Matrix

Defaults

Micylou builds a start-up solution based on the original Mediawiki LTS package and includes additional extensions to build its start-up solution.

Therefore, the roles and rights included in these extensions are taken into account, even if the extension is not activated.

Roles by groups

Source	Permission	Description	By default	As from	* Reset	* Granted	User Reset	User Granted	sysop Granted	Bot Granted	Bureaucrat Granted	Linux-Admin Granted	Wiki-Admin Granted	Wiki-Moderator Granted	Wiki-Editor Granted	Wiki-Updater Granted	
Default Mediawiki	apihighlimits	Use higher limits in API queries	bot, sysop	1.12+													
Default Mediawiki	applychangetags	Apply tags along with one's changes	user	1.25+			X		X	X						X	
Default Mediawiki	autoconfirmed	Not be affected by IP-based rate limits - used for the 'autoconfirmed' group, see the other table below for more information	autoconfirmed, bot, sysop	1.6+													
Default Mediawiki	autocreateaccount	Automatically log in with an external user account - a more limited version of createaccount	—	1.27+		X		LDAP REQUIREMENT									
Default Mediawiki	autopatrol	Have one's own edits automatically marked as patrolled - $wgUseRCPatrol must be true	bot, sysop	1.9+										X	X		
Default Mediawiki	bigdelete	Delete pages with large histories (as determined by $wgDeleteRevisionsLimit)	sysop	1.12+						X				X			
Default Mediawiki	block	Block other users from editing - Block options include preventing editing and registering new accounts, and autoblocking other users on the same IP address	sysop	1.5+						X			X	X			
Default Mediawiki	blockemail	Block a user from sending email - allows preventing use of the Special:Emailuser interface when blocking - requires the block right	sysop	1.11+						X			X	X	X		
Default Mediawiki	bot	Be treated as an automated process - can optionally be viewed	bot	1.5+													
Default Mediawiki	browsearchive	Search deleted pages - through Special:Undelete	sysop	1.13+					X	X				X	X	X	
Default Mediawiki	changetags	Add and remove arbitrary tags on individual revisions and log entries - currently unused by extensions	user	1.25+	X		X		X	X				X	X	X	
Default Mediawiki	createaccount	Create new user accounts - register / registration	*, sysop	1.5+	X				X	X		X		X			
Default Mediawiki	createpage	Create pages (which are not discussion pages) - requires the edit right	*, user	1.6+	X		X		X	X				X	X	X	X
Default Mediawiki	createtalk	Create discussion pages - requires the edit right	*, user	1.6+	X		X							X			
Default Mediawiki	delete	allows the deletion or undeletion of pages.	sysop	1.5+					X	X			X	X	X	X	X
Default Mediawiki	deletechangetags	Delete tags from the database - currently unused by extensions	sysop	1.28+										X			
Default Mediawiki	deletedhistory	View deleted history entries, without their associated text	sysop	1.6+					X	X				X	X	X	X
Default Mediawiki	deletedtext	View deleted text and changes between deleted revisions	sysop						X	X				X	X	X	X
Default Mediawiki	deletelogentry	Delete and undelete specific log entries - allows deleting/undeleting information (action text, summary, user who made the action) of specific log entries - requires the deleterevision right (not available by default)	sysop	1.20+					X	X							

Default Mediawiki	deleterevision	Delete and undelete specific revisions of pages - allows deleting/undeleting information (revision text, edit summary, user who made the edit) of specific revisions Split into deleterevision and deletelogentry in 1.20 (not available by default)	sysop	1.6+					X	X				X	X	X	
Default Mediawiki	edit	Edit pages	*, user	1.5+	X	X			X	X						X	X
Default Mediawiki	editcontentmodel	Edit the content model of a page	user	1.23.7+	X	X			X	X				X	X	X	
Default Mediawiki	editinterface	Edit the user interface - contains interface messages. For editing sitewide CSS/JSON/JS, there are now segregate rights, see below.	sysop, interface-admin	1.5+					X	X				X			
Default Mediawiki	editmyoptions	Edit your own preferences	*	1.22+	X				X	X				X	X	X	X
Default Mediawiki	editmyprivateinfo	Edit your own private data (e.g. email address, real name)	*	1.22+	X				X	X				X	X	X	X
Default Mediawiki	editmyusercss	Edit your own user CSS files - prior to 1.31 it was assigned to everyone (i.e. "*") (note that this is not needed if the group already has the editusercss right)	user	1.22+			X										
Default Mediawiki	editmyuserjs	Edit your own user JavaScript files - prior to 1.31 it was assigned to everyone (i.e. "*") (note that this is not needed if the group already has the edituserjs right)	user	1.22+			X										
Default Mediawiki	editmyuserjson	Edit your own user JSON files (note that this is not needed if the group already has the edituserjson right)	user	1.31+			X										
Default Mediawiki	editmywatchlist	Edit your own watchlist. Note some actions will still add pages even without this right.	*	1.22+			X										
Default Mediawiki	editprotected	Edit pages protected as "Allow only administrators" - without cascading protection	sysop	1.13+					X	X				X	X	X	
Default Mediawiki	editsemiprotected	Edit pages protected as "Allow only autoconfirmed users" - without cascading protection	autoconfirmed, bot, sysop	1.22+					X	X				X	X	X	
Default Mediawiki	editsitecss	Edit sitewide CSS	interface-admin	1.32+					X	X				X			
Default Mediawiki	editsitejs	Edit sitewide JavaScript	interface-admin	1.32+					X	X				X			
Default Mediawiki	editsitejson	Edit sitewide JSON	sysop, interface-admin	1.32+					X	X				X			
Default Mediawiki	edituser css	Edit other users' CSS files	interface-admin	1.16+					X	X				X			
Default Mediawiki	edituserjs	Edit other users' JavaScript files	interface-admin	1.16+					X	X				X			
Default Mediawiki	edituserjson	Edit other users' JSON files	sysop, interface-admin	1.31+					X	X				X			
Default Mediawiki	hideuser	Block a username, hiding it from the public - (not available by default)	—	1.10+					X	X				X	X		
Default Mediawiki	import	Import pages from other wikis - “transwiki”	sysop	1.5+					X	X				X		X	
Default Mediawiki	importupload	Import pages from a file upload - This right was called 'importraw' in and before version 1.5	sysop	1.5+					X	X				X			
Default Mediawiki	ipblock-exempt	Bypass IP blocks, auto-blocks and range blocks	sysop	1.9+					X	X				X			
Default Mediawiki	managechangetags	Create and (de)activate tags - currently unused by extensions	sysop	1.25+					X	X				X			
Default Mediawiki	markbotedits	Mark rolled-back edits as bot edits - see Manual:Administrators#Rollback	sysop	1.12+					X	X				X			
Default Mediawiki	mergehistory	Merge the history of pages	sysop	1.12+					X	X				X			
Default Mediawiki	minoredit	Mark edits as minor	user	1.6+		X			X	X				X		X	X
Default Mediawiki	move	Move pages - requires the edit right	user, sysop	1.5+		X			X	X				X		X	X
Default Mediawiki	move-categorypages	Move category pages - requires the move right	user, sysop	1.25+		X			X	X				X		X	
Default Mediawiki	movefile	Move files - requires the move right and $wgAllowImageMoving to be true	user, sysop	1.14+		X			X	X				X		X	
Default Mediawiki	move-rootuserpages	Move root user pages - requires the move right	user, sysop	1.14+		X			X	X				X		X	
Default Mediawiki	move-subpages	Move pages with their subpages - requires the move right	user, sysop	1.13+		X			X	X				X		X	
Default Mediawiki	nominornewtalk	Not have minor edits to discussion pages trigger the new messages prompt - requires the minor edit right	bot	1.9+													
Default Mediawiki	noratelimit	Not be affected by rate limits - not affected by rate limits (prior to the introduction of this right, the configuration variable $wgRateLimitsExcludedGroups was used for this purpose)	sysop, bureaucrat	1.13+					X	X		X		X			
Default Mediawiki	override-export-depth	Export pages including linked pages up to a depth of 5	—	?	X				X	X				X			
Default Mediawiki	pagelang	Change page language - $wgPageLanguageUseDB must be true	—	1.24+	X				X	X				X			
Default Mediawiki	patrol	Mark others' edits as patrolled - $wgUseRCPatrol must be true	sysop	1.5+					X	X				X			
Default Mediawiki	patrolmarks	View recent changes patrol marks	—	1.16+	X				X	X				X	X		
Default Mediawiki	protect	Change protection levels and edit cascade-protected pages	sysop	1.5+					X	X						X	X
Default Mediawiki	purge	Purge the site cache for a page - URL parameter "&action=purge"	user	1.10+		X			X	X				X			
Default Mediawiki	read	Read pages - when set to false, override for specific pages with $wgWhitelistRead		1.5+	X		X										
Default Mediawiki	readapi		*, user, bot	1.13+	X	X			X	X	X	X	X	X	X	X	X
Default Mediawiki	reupload	Overwrite existing files - requires the upload right	user, sysop	1.6+		X			X	X				X	X	X	X
Default Mediawiki	reupload-own	Overwrite existing files uploaded by oneself - requires the upload right (note that this is not needed if the group already has the reupload right)	—	1.11+	X				X	X				X	X	X	X
Default Mediawiki	reupload-shared	Override files on the shared media repository locally - (if one is set up) with local files (requires the upload right)	user, sysop	1.6+		X			X	X				X	X	X	X
Default Mediawiki	rollback	Quickly rollback the edits of the last user who edited a particular page	sysop	1.5+					X	X				X	X		
Default Mediawiki	sendemail	Send email to other users	user	1.16+		X			X	X		X	X	X	X	X	X
Default Mediawiki	siteadmin	Lock and unlock the database - which blocks all interactions with the web site except viewing. (not available by default)	—	1.5+					X	X			X	X			
Default Mediawiki	suppressionlog	View private logs	—	1.6+					X	X			X	X			
Default Mediawiki	suppressredirect	Not create redirects from source pages when moving pages	bot, sysop	1.12+					X	X				X			
Default Mediawiki	suppressrevision	View, hide and unhide specific revisions of pages from any user - Prior to 1.13 this right was named hiderevision (not available by default)	—	1.6+					X	X				X			
Default Mediawiki	unblockself	Unblock oneself - Without it, an administrator that has the capability to block cannot unblock themselves if blocked by another administrator	sysop	1.17+					X	X				X			
Default Mediawiki	undelete	Undelete a page - requires the deletedhistory right	sysop	1.12+					X	X				X	X	X	

Default Mediawiki	unwatchedpages	View a list of unwatched pages - lists pages that no user has watchlisted	sysop	1.6+					X	X					X			
Default Mediawiki	upload	Upload files - requires the edit right and $wgEnableUploads to be true	user, sysop	1.5+		X		X	X		X	X	X		X	X	X
Default Mediawiki	upload_by_url	Upload files from a URL - requires the upload right (Prior to 1.20 it was given to sysops)	—	1.8+					X			X	X				
Default Mediawiki	userrights	Edit all user rights - allows the assignment or removal of all* groups to any user.	bureaucrat	1.5+				X	X		X		X				
Default Mediawiki	userrights-interwiki	Edit user rights of users on other wikis	—	1.12+					X			X	X				
Default Mediawiki	viewmyprivateinfo	View your own private data (e.g. email address, real name)	*	1.22+	X		X										
Default Mediawiki	viewmywatchlist	View your own watchlist	*	1.22+	X		X										
Default Mediawiki	viewsuppressed	View revisions hidden from any user - i.e. a more narrow alternative to "suppressrevision" (note that this is not needed if the group already has the suppressrevision right)	—	1.24+				X	X							X	
Default Mediawiki	writeapi	Use of the write API	*, user, bot	1.13+	X	X		X	X	X	X	X	X	X	X	X
ExtensionRevsApprove	$egApprovedRevsBlankIfUnapproved		FALSE														
ExtensionRevsApprove	$egApprovedRevsShowApproveLatest		TRUE														
ExtensionRevsApprove	$egApprovedRevsShowNotApprovedMessage		TRUE														
ExtensionRevsApprove	approverevision				X	X		X	X						X		
ExtensionRevsApprove	egApprovedRevsAuto$maticApprovals		FALSE														
ExtensionRevsApprove	egApprovedRevsBlan$kIfUnapproved		FALSE														
ExtensionRevsApprove	egApprovedRevsShowNotApprovedMessa$ge		TRUE														
ExtensionRevsApprove	viewapprover				X	X	X	X	X	X	X	X	X	X	X	X
ExtensionRevsApprove	viewlinktolatest				X	X								X	X	X	X
Extention UserMerge	usermerge							X	X		X			X			
Extension EditUser	edituser							X	X		X						

[Back to top of this page](#)

[Back to Welcome Page](#)

Security matrix of our default solution set up in LocalSettings.php - restricted access

View of the detailed Mediawiki Security Configuration is restricted.

[Back to top of this page](#)

[Back to Welcome Page](#)

Mediawiki Toolbox Personnalisation

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Configuration
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Remove the Sidebar and main logo at MediaWiki:Sidebar To completely remove the sidebar and logo:

Blank the MediaWiki:Sidebar page and save.

Then add:

```
#column-content { margin: 0 0 1em 0; }
#content { margin: 0 0 0 0; }
#p-cactions { left: .1em; }
```

This moves the main text of your wiki to the left.

Add or remove toolbox sections (JavaScript) The toolbox appears on the left under MediaWiki:Sidebar. The toolbox is a dynamic element requiring programmatic methods such as skin extensions, PHP (see skins/MonoBook.php and creating a new skin) or JavaScript.

The JavaScript solution uses User:{username}/common.js / MediaWiki:Common.js, available for MediaWiki 1.9+.

Entire MediaWiki website: copy the script below into MediaWiki:Common.js
One specific user: copy the script below into Special:MyPage/common.js (or the JS page for the user's current skin, such as Special:MyPage/monobook.js).

Now simply configure which link should appear in which section. You also can remove some links if you want to.

```
function ModifySidebar( action, section, name, link ) {
        try {
                switch ( section ) {
                        case 'languages':
                                var target = 'p-lang';
                                break;
                        case 'toolbox':
                                var target = 'p-tb';
                                break;
                        case 'navigation':
                                var target = 'p-navigation';
                                break;
                        default:
                                var target = 'p-' + section;
                                break;
                }

                if ( action == 'add' ) {
                        var node = document.getElementById( target )
.getElementsByTagName( 'div' )[0]
.getElementsByTagName( 'ul' )[0];

                        var aNode = document.createElement( 'a' );
                        var liNode = document.createElement( 'li' );

                        aNode.appendChild( document.createTextNode( name ) );
                        aNode.setAttribute( 'href', link );
                        liNode.appendChild( aNode );
```

```
                        liNode.className = 'plainlinks';
                        node.appendChild( liNode );
                }

                if ( action == 'remove' ) {
                        var list = document.getElementById( target )
.getElementsByTagName( 'div' )[0]
.getElementsByTagName( 'ul' )[0];

                        var listelements = list.getElementsByTagName( 'li' );

                        for ( var i = 0; i < listelements.length; i++ ) {
                                if (
                                        listelements[i].getElementsByTagName(
'a' )[0].innerHTML == name ||
                                        listelements[i].getElementsByTagName(
'a' )[0].href == link
                                )
                                {
                                        list.removeChild( listelements[i] );
                                }
                        }
                }


        } catch( e ) {
                // let's just ignore what's happened
                return;
        }
}

function CustomizeModificationsOfSidebar() {
        // adds [[Special:CategoryTree|Special:CategoryTree]] to toolbox
        ModifySidebar( 'add', 'toolbox', 'CategoryTree',
'https://en.wikipedia.org/wiki/Special:CategoryTree' );
        // removes [[Special:Upload|Special:Upload]] from toolbox
        ModifySidebar( 'remove', 'toolbox', 'Upload file',
'https://en.wikipedia.org/wiki/Special:Upload' );
}

jQuery( CustomizeModificationsOfSidebar );
```

Mediawiki backup script

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Linux

Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	Micylou restricted

#!/bin/bash # # wikigial fullsitebackup.sh V1.0 # # Full backup of website files and database content. # # - Added mysql password variable (caution - this script file is now a security risk - protect it) # - Generates temp log file # - Updated backup and restore scripts have been tested on Ubunutu Edgy server w/Drupal 5.1 # # # Parameters: # tar_file_name (optional) # ################# # Configuration # ## ## UTM VM Database connection information ### dbname="my_wiki" # name of the database dbhost="localhost" # database server dbuser="wikidbadmin" # database user dbpw="Gyzmo6522!" # database user password dbport="3306" # database port webrootdir="/var/www/html/wiki/" ### folder where mediawiki root folder is locater startdir="/var/www/html/backups/" ### backup file location tarnamebase=sitebackup- datestamp=`date +'%Y%m%d'-'%H%M%S'` # Execution directory (script start point) logfile=$startdir"backup.log" # file path and name of log file to use # Temporary Directory tempdir=$datestamp ## ### ################ # Start menu ############### clear select delconf in "Backup" "Cancel" do echo "You have selected :" $delconf case $delconf in Backup) echo "Proceeding with Mediawiki full site backup"; break;; Cancel) exit;; esac break done ################# # Website Files ################# cd $webrootdir echo "SELECT THE VERSION TO BACKUP (ctrl-c to quit) :" echo "" select mediawikidir in mediawiki-*.*; do echo "" echo "You have selected $mediawikidir ($REPLY)." break done ########################### # Input Parameter Check # ########################### echo "Input parameters check" if test "$1" = "" then tarname=$tarnamebase$datestamp else tarname=$1 fi echo "Input parameters check" ################# # Begin logging # ################# echo "Beginning backup" echo "Beginning $mediawikidir site backup using FullSiteBackup.sh ..." > $logfile ## # Create temporary working directory # ## echo " Creating temp working temporary directory" echo " Creating temp working dir ..." >> $logfile mkdir $startdir/$tempdir ###################### # ZIP website files # ###################### echo " TARing website files..." echo " TARing website files into $webrootdir ..." >> $logfile cd $webrootdir sudo tar czvf $startdir/$tempdir/filecontent.tar.gz $mediawikidir ################################## # sqldump database information # ################################## echo " Dumping mediawiki database ..." >> $logfile ## echo " Dumping mediawiki database, using ..." >> $logfile ## echo " user:$dbuser; database:$dbname host:$dbhost " >> $logfile cd $startdir/$tempdir mysqldump --host=$dbhost -P$dbport --user=$dbuser --password=$dbpw --add-drop-table $dbname > dbcontent.sql clear ############################## # Create final backup file # # ########################### echo "Create final compressed tgz file..." echo " Creating final compressed (tgz) TAR file: $tarname$mediawikidir" >> $logfile sudo tar -czf $startdir/$tarname-$mediawikidir.tar.gz filecontent.tar.gz dbcontent.sql ########### # Cleanup # ########### echo " Removing temporary directory ..." echo " Removing temp dir $tempdir ..." >> $logfile cd $startdir rm -r $tempdir echo "$temdir directory removed " echo

"Backup complete, have a nice day." echo ############### # Exit banner # # #############

Syntaxe Mediawiki - How-to ?

Mediawiki Extension

Source:	DataSource
Language:	Français
Topic:	Mediawiki
SubTopic:	Syntax
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	User Guide
Status:	Active
Access:	free

Download this page as PDF

Cet documentvous présente la **syntaxe wiki**, c'est-à-dire la façon de mettre en forme le texte, de créer des liens, etc. propre à MediaWiki. Cette syntaxe reprend en partie celle de l'HTML. Il résume sous forme de tableaux les différentes mises en forme possibles. La colonne de droite indique les codes à saisir, et celle de gauche l'effet obtenu.

Attention, Mediawiki est installé en anglais => toutes les commandes sont à coder dans la langue sauce menthe.

Table des matières

Une table des matières est créée automatiquement si l'article comporte plus de 3 titres ou sous-titres (au moins 4).

Pour **supprimer le sommaire automatique**, il suffit de mettre quelque part __NOTOC__.

Pour **modifier l'emplacement de la table des matières**, il suffit de placer __NOTOC__ en haut de la page, puis d'insérer __TOC__ là où vous désirez voir apparaître la table.

Il n'existe pour le moment aucune façon de limiter la profondeur des tables des matières.

Mise en page

Chapitres, sections, paragraphes

Pour les titres :
  Nouveau chapitre   Nouvelle section   Nouvelle sous-section   Nouveau paragraphe   *Les notations ci-contre sont respectivement équivalentes à l'utilisation des balises html ... , ... , ... et ... .*
De plus, chaque titre va être ajouté dans le sommaire qui est généralement situé en haut de la page.
Souvent, les sections ne commencent qu'au niveau 2, puisque le niveau 1 est comme le titre.
	`= Nouveau chapitre =`
	`== Nouvelle section ==`
	`=== Nouvelle sous-section ===`
	`==== Nouveau paragraphe ====`

Retour à la ligne

Un simple retour à la ligne n'a aucun effet.
 	
Mais une ligne vide
permet de séparer les paragraphes.
	`Un simple retour à la ligne`
	`n'a aucun effet.`
	
	`Mais une ligne vide`
	
	`AVEC UNE AUTRE LIGNE VIDE`
	
	`permet de séparer les paragraphes.`

Voilà comment il faut faire pour aller à la ligne. *Note* : la balise est suffisante.	`Voilà comment il faut faire ` `pour aller à la ligne.`

Les listes

• Les listes sont bien pratiques :
- Elles permettent d'organiser les données
- Elles embellissent le document
	`* Les listes sont bien pratiques :`
	`** Elles permettent d'organiser les données`
	`** Elles embellissent le document`

1. Les listes numérotées sont également bien pratiques
2. Bien organisées
1. Faciles à lire
	`# Les listes numérotées sont également bien pratiques`
	`# Bien organisées`
	`## Faciles à lire`

Les lignes
peuvent être
indentées
sur plusieurs niveaux
	`Les lignes`
	`:peuvent être`
	`::indentées`
	`:::sur plusieurs niveaux`

Avec tout cela, il est maintenant possible de mélanger les types de listes :
• Une liste numérotée :
1. Premier élément
2. Second élément
• d'autres branches à l'arborescence
	`*Une liste numérotée :`
	`:#Premier élément`
	`:#Second élément`
	`*d'autres branches à l'arborescence`

Listes de définitions listes contenant des définitions Terme à expliquer explication du terme	`; Listes de définitions : listes contenant des définitions` `; Terme à expliquer : explication du terme`

En utilisant des balises (*tags*) HTML c'est plus beau (mais c'est plus difficile à mémoriser) :
• Une liste numérotée :
1. Premier élément
2. Second élément
• d'autres branches à l'arborescence
Petit truc mnémotechnique :
• *ol = Liste Ordonnée (c'est à dire numérotée - Ordered List en anglais)* ;
• *ul = Liste "Unordonnée" (Unordered List en anglais) ou plus prosaïquement liste à puces* ;
• *li = élément d'une liste (List Item en anglais).*
	`<ul>`
	`  <li>Une liste numérotée :</li>`
	`  <ol>`
	`  <li>Premier élément</li>`
	`  <li>Second élément</li>`
	`  </ol>`
	`  <li>d'autres branches à l'arborescence</li>`
	`</ul>`

Effets divers

SI une ligne commence par une espace ALORS elle apparaîtra exactement telle qu'elle a été saisie ; dans une police non proportionnelle ; les sauts de ligne ne seront pas automatiques ; FIN SI C'est bien utile pour : • copier/coller un texte pré-formaté ; • des descriptions d'algorithmes ; *Note : vous forcez ainsi la page à être relativement large, ce qui peut être gênant à lire.*	SI (une ligne commence par une espace) ALORS elle apparaîtra exactement telle qu'elle a été saisie ; dans une police non proportionnelle; les sauts de ligne ne seront pas automatiques ; FIN SI C'est bien utile pour : • copier/coller un texte pré-formaté ; • des descriptions d'algorithmes ;
Le texte peut être centré.	`<center>Le texte peut être centré.</center>`
Une ligne horizontale : au-dessus --- et en-dessous.	Une ligne horizontale : au-dessus ---- et en-dessous.

Mise en forme des caractères

Faire ressortir un mot, **un peu plus**, ***encore plus***. À utiliser également dans les formules : **F** = *m* **a**.	''Faire ressortir un mot'', '''un peu plus''', '''''encore plus'''''. '''F''' = ''m'' '''a'''.
Texte en couleur	`<span style="color:#FF0000;">Texte en couleur</span>`
Différentes tailles de polices : petite taille grande taille	`<small>petite taille</small>` `<big>grande taille</big>`
Police machine à écrire : `termes techniques`.	`<tt>termes techniques</tt>`

Vous pouvez ~~barrer~~ *certains mots et* en souligner d'autres...	`<strike>barrer</strike>` `<u>en souligner d'autres...</u>`

Umlauts et accents :
À Á Â Ã Ä Å Æ
Ç È É Ê Ë Ì Í
Î Ï Ñ
Ò Ó Ô Õ Œ
Ö Ø Ù Ú Û Ü ß
à á â ã ä å æ
ç è é ê ë ì í
î ï ñ ò ó ô õ œ
ö ø ù ú û ü ÿ
Note : depuis le passage à l'Unicode on peut aussi parfaitement écrire les caractères spéciaux directement sans aucun souci.
	`À Á Â Ã Ä`
	`Å Æ`
	`Ç È É Ê Ë`
	`Ì Í`
	`Î Ï Ñ Ò Ó`
	`Ô Õ Œ`
	`Ö Ø Ù Ú Û`
	`Ü ß`
	`à á â ã ä`
	`å æ`
	`ç è é ê ë`
	`ì í`
	`î ï ñ ò ó`
	`ô õ œ`
	`ö ø ù ú û`
	`ü ÿ`

Ponctuations :
¿ ¡ « » § ¶ † ‡ •
	`¿ ¡ « » §`
	`¶`
	`† ‡ •`

Caractères spéciaux : ™ © ® ¢ € ¥ £ ¤	`™ © ® ¢ € ¥` `£ ¤`
Indice (subscript en anglais) : x_2 *et* *Exposant (superscript en anglais) :* x^2	`Indice : x<sub>2</sub> et` `Exposant : x<sup>2</sup>`

Caractères grecs :
α β γ δ ε ζ η
θ ι κ λ μ ν ξ
π ρ σ ς τ υ φ
χ ψ ω
Γ Δ Θ Λ Ξ Π Σ
Φ Ψ Ω
	`α β γ δ ε`
	`ζ η`
	`θ ι κ λ μ ν`
	`ξ`
	`π ρ  σ ς τ υ`
	`φ`
	`χ ψ ω`
	`Γ Δ Θ Λ Ξ Π`
	`Σ`
	`Φ Ψ Ω`

Caractères mathématiques :
∫ ∑ ∏ √ ± ∞
≈ ∝ ≡ ≠ ≤ ≥ →
× · ÷ ∂ ′ ″ ∇
‰ ° □ □
ø ∈ ∋ ∩ ∪ ⊂ ⊃ ⊆ ⊇
¬ ∧ ∨ ∃ ∀ ⇒ ⇔
	`∫ ∑ ∏ √ ± ∞`
	`≈ ∝ ≡ ≠ ≤ ≥ →`
	`× · ÷ ∂ ′`
	`″ ∇`
	`‰ ° ∴ ℵ`
	`ø ∈ ∋ ∩ ∪ ⊂ ⊃`
	`⊆ ⊇`
	`¬ ∧ ∨ ∃ ∀ ⇒`
	`⇔`

Plus d'espace que d'habitude, sans saut de ligne : $x^2 \geq 0$ est vrai. Utile pour les formules.	`x<sup>2</sup>   ≥   0`

ASCII art (Commencer chaque ligne par un blanc) :
∞ -x²
∫ e dx = √π
-∞
Malgré tout, la racine carrée est très moche.
	`∞ -x<sup>2</sup>`
	` ∫ e  dx = √π`
	` -∞`

Formules mathématiques compliquées : `<math>\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}</math>`	`<math>\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}</math>`

Suppression de l'interprétation d'une balise : Lien → (vers) l'[[Aide:Contenu|Index]] *Les balises ... et [[...]] apparaissent en clair. Elles ne sont pas interpretées par le logiciel.* *• Est utilisé pour montrer littéralement des données qui autrement auraient une signification spéciale.* *• Toutes les* balises *wiki sont ignorées, y compris celles ressemblant à des tags HTML.* *• Les caractères HTML spéciaux sont toujours interprétés.*	`<nowiki>Lien → (<i>vers</i>) la [[Wikipédia:FAQ]]</nowiki>`

Liens et adresses internet

Les liens

Liens internes

Lien vers Accueil.	`Lien vers [[Main_Page|Accueil]].`
page d'accueil *C'est le même lien que précédemment mais avec une étiquette différente.* On parle également de lien redirigé	`[[Accueil|page d'accueil]]`
La sexualité des mouches dans le Vercors oriental. *C'est une page qui n'existe pas encore.* *Vous pouvez la créer en cliquant simplement sur le lien.* *Après l'avoir créée, faites une recherche sur son titre, et assurez-vous que les autres articles y font correctement référence.*	`[[La sexualité des mouches dans le Vercors oriental]].`
Pour référencer les pages utilisateur, il existe un raccourci pour éviter de taper deux fois le nom : DochyJP est équivalent à DochyJP	`[[User:DochyJP||]] est équivalent à [[User:DochyJP|DochyJP]]`
Il est possible de pointer directement vers un paragraphe voire un sous-paragraphe d'un article donné: il suffit de faire suivre le nom de l'article de la touche # et du titre du paragraphe ou sous paragraphe en respectant l'orthographe, majuscules comprises *Le lien suivant :* Mon lien *pointe directement vers le paragraphe des équipes GIAL dans le manuel de formation des nouveeaux collaborateurs du Service Desk.*	`[[Manuel_de_formation_d'un_nouveau_collaborateur_du_Service_Desk#Les différentes équipes de GIAL]]`

Liens externes

Lien vers http://www.google.fr . *Il suffit d'écrire l'adresse en entier pour qu'un lien soit créé*	`Lien vers http://www.google.fr`
Suivre ce lien : [1]. *Un lien externe s'écrit entre crochets simples.* *Il est remplacé automatiquement par un numéro.*	`Suivre ce lien : [http://www.google.fr].`

Aller sur le site de Google. *Pour écrire une étiquette, il suffit de laisser un espace après l'adresse.* *Tout le texte après l'espace constitue l'étiquette, qui peut elle-même contenir des espaces :* Google, le moteur de recherche	`Aller sur le site de [http://www.google.fr Google].` `[http://www.google.fr Google, le moteur de recherche].`

Les points d'ancrage

Les points d'ancrage :
ancre 1.
ancre 2.
Ces deux lignes sont des points d'ancrage vers lesquels des liens internes ou externes à cette page peuvent pointer.
	`<span id="ancre_1">ancre 1.</span>`
	
	`<span id="ancre_2">ancre 2.</span>`
	Ces deux points d'ancrage sont utilisés comme exemple dans la page Ancres.

Les redirections

La redirection vers l'accueil d'une page vers une autre est possible en mettant ce texte dans la première ligne : *L'utilisation de France comme redirection est, bien sûr, un exemple.*	`#Main_Page|redirection vers l'accueil par ex`

Les adresses internet

Elles sont également appelées *liens externes*. Voyez aussi la page Lien.

Lien externe : Wikipédia *Lorsque vous imprimez la page, une autre feuille de style est utilisée et l'adresse apparaîtra clairement sur le papier.*	`Lien externe : [http://fr.wikipedia.org Wikipédia]`
Mais l'adresse internet suffit : http://fr.wikipedia.org.	`Mais l'adresse internet suffit : http://fr.wikipedia.org .`

Les images

Il n'est possible d'utiliser que des images qui ont été téléchargées vers le serveur.
Voyez Images et fichiers importés pour le chargement d'un fichier sur le serveur et pour une explication complète pour l'utilisation de l'image.

Syntaxe

[[File:filename.extension|options|caption]]
ou "options" peut être vide ou bien est composé des éléments suivants, séparés par le symbole tube (pipe) "|"

- Option du format : un des bordure et/ou sans cadre, avec cadre, miniature; si plusieurs sont présents, seul le premier est utilisé;

Contrôle la façon dont l'image affichée est formatée et incluse dans le reste de la page.

- Option de redimensionement : un de
  - {width}**px** — Redimensionne l'image pour l'adapter dans la largeur maximale donnée en pixels, sans restreindre sa hauteur; (A space character between the width value and "px" is permitted.)
  - **x**{height}**px** — Redimensionne l'image pour l'adapter à une hauteur maximale donnée en pixels, sans restreindre sa largeur;
  - {width}**x**{height}**px** — Redimensionner l'image pour l'adapter à une largeur et une hauteur données en pixels;
  - **upright** — Redimensionne une image pour l'adapter à des dimensions raisonnables, selon les préférences de l'utilisateur (adapté aux images dont la taille est supérieure à la largeur). Utiliser |upright=1.0| montrera l'image à la taille par défaut; |upright=2.0| la montrera au double de la taille par défaut. La taille maximale par défaut dépend du format et des dimensions de l'image interne (en fonction de son type de support)
  - **Option Horizontal alignment** : parmi **left** (gauche), **right** (droite), **center** (centré), **none** (aucun); si plusieurs sont présents, seul le premier est utilisé; cela contrôle l'alignement horizontal (et en ligne / bloc ou styles flottants) de l'image dans un texte (pas de valeur par défaut).
  - **Option Vertical alignment** : parmi **baseline** (line de base), **sub** (sous), **super**, **top** (haut), **text-top** (haut du texte), **middle** (milieu), **bottom** (bas), **text-bottom** (bas du texte) : cela contrôle l'alignement vertical d'une image non-flottante en ligne avec le texte avant ou après l'image, et dans le même bloc (l'alignement vertical par défaut est 'milieu').
  - Option Link: parmi
    - **link=**{cible} — Permet de changer la cible (vers un titre de page arbitraire, ou une URL) du lien généré, activable pour l'affichage de l'image; à savoir, affichage [[File:Exemple.jpg|20px|link=http://www.wikipedia.ord]] interprété comme Exemple.jpg (lien externe), ou [[File:MediaWiki logo embroidery.PNG|40px|Exemple.PNG]] interprété comme (lien interne).
    - link= (valeur vide) Affiche une image sans aucun lien activable; à savoir [[File:MediaWiki logo embroidery.PNG|20px|link=]] interprété comme

.

- Autre option spécifiques :
    - **alt=**{texte alternatif} — (MediaWiki 1.14+) Définit le texte alternatif (attribut HTML alt = "..." de l'élément d'une image <img />) généré qui sera affiché soit si l'image référencée ne peut pas être téléchargé et/ou intégré, soit si le support du média nécessite l'utilisation du texte de description de remplacement (par exemple lors de l'utilisation d'un lecteur Braille ou avec des options d'accessibilité définies par l'utilisateur dans son navigateur).
    - **page=**{nombre} : Affiche le numéro de page spécifié (actuellement seulement applicable lors de l'affichage d'un fichier .djvu ou .pdf).
    - **class=**{class html} : Définit les classes (attribut HTML class="..." de l'élément <img /> généré).
    - **lang=**{language code} : Pour les fichier SVG contenant <switch> des instructions variables selon l'attribut langue système, sélectionne quelle langue pour afficher le fichier.

Si un paramètre ne correspond à aucune de ces possibilités, il est interprété comme étant le texte de la légende de l'image. If more than one non-parameter string is present, the final non-parameter string will be used as the caption. La légende s'affiche en dessous de l'image pour les formats "thumb" et "frame", pour les autres c'est une infobulle. Dans le cas "thumb" ou "frame", la légende peut contenir des liens wiki ou autre marques de formatage. Des options supplémentaires peuvent êtres disponibles selon les extensions MediaWiki installées.

Si l'option "alt" n'est pas renseignée, et qu'une légende est présente, le texte alternatif (le champ alt de l'élément HTML img) sera créé automatiquement à partir de la légende, avec les marques de formatage automatiquement supprimées, sauf en mode vignette ou cadre car la légende est déjà lisible par les lecteurs d'écran dans ce cas.

Tableaux

Les tableaux peuvent être très utiles pour la représentation de certaines données sur Wikipédia. Si vous êtes déjà familier des codes HTML permettant de créer des tableaux, vous pouvez insérer simplement le code directement dans l'article que vous êtes en train d'éditer.

Cependant, les balises de tableaux sont parfois difficiles à éditer surtout pour quelqu'un n'étant pas à l'aise avec la codification HTML. Pour cela, des nouvelles balises développées par Magnus Manske ont été créées. Ces balises remplacent les traditionnelles balises HTML <table>, <tr>, <td>, <th> et <caption>. On peut donc se passer totalement des balises HTML ; il est même plutôt conseillé de ne plus utiliser de code HTML pour créer un tableau, bien que son utilisation soit encore possible.

L'utilisation de tableaux doit être évitée si cela est possible. En effet, l'accessibilité aux pages qui contiennent un tableau est rendue plus difficile à certaines personnes, notamment aux aveugles ou aux débiles.

Ce petit outil XL permet de transformer 19 colones sur 100 lignes en format wiki : File:Tableau XL vers tableau WIKI.xlsx

Petits exemples

Ici, deux exemples simples pour avoir un premier contact avec le codage Wiki. Notre premier exemple nous montrera le tableau le plus minimaliste qui soit, c'est à dire celui qui ne contiendrait qu'une seule cellule.

Résultat affiché	Codage Wiki	Codage HTML

cellule	`{| border="1"`
	`|cellule`
	`|}`
		`<table border="1">`
		`  <tr>`
		`  <td>cellule</td>`
		`  </tr>`
		`</table>`

Le deuxième exemple contient deux cellules. Comme vous pouvez le constater, il est possible de disposer les cellules de deux façons différentes. Soit les cellules sont placées les unes en dessous des autres, soit elles sont placées les unes à côté des autres. Mais nous y reviendrons plus en détail par la suite.

Résultat affiché	Codage Wiki	Codage HTML

gauche droite	`{| border="1"`
	`|gauche`
	`|droite`
	`|}`
		`<table border="1">`
		`  <tr>`
		`  <td>gauche</td>`
		`  <td>droite</td>`
		`  </tr>`
		`</table>`

gauche droite	`{| border="1"`
	`|gauche || droite`
	`|}`
		`<table border="1">`
		`  <tr>`
		`  <td>gauche</td>`
		`  <td>droite</td>`
		`  </tr>`
		`</table>`

La syntaxe des tableaux

Nous verrons ici les différentes balises qui peuvent apparaître dans la création d'un tableau.

table

Dans le langage HTML, l'élément `table` est l'élément de base pour la création d'un tableau. En wiki, pour créer un tableau, nous procédons de la façon suivante :

```
{| paramètres
 |}
```

ce qui donnerait en HTML, l'encodage suivant :

```
<table paramètres>
</table>
```

On voit donc bien qu'un tableau commence toujours par l'accolade ouvrante **{**, suivie d'un trait vertical |. La fin d'un tableau est également composée de deux caractères. Le premier est le trait vertical | et il est suivi du deuxième caractère qui est l'accolade fermante **}**. Le champ *paramètres* doit bien entendu être remplacé par différents paramètres utilisés dans les tableaux tel que la couleur de fond, la taille, s'il faut faire apparaître une bordure, etc. L'utilisation des paramètres n'est pas obligatoire et on peut donc sans problème ne rien placer à cet endroit.

td

En HTML, l'élément `td` représente une cellule dans une ligne du tableau. Pour créer des cellules, la syntaxe est la suivante :

```
|cellule1
|cellule2
|cellule3
```

Une autre disposition des cellules totalement équivalente est la suivante :

```
|cellule1||cellule2||cellule3
```

Cette deuxième disposition impose qu'entre deux cellules apparaissent deux traits verticaux || car un seul trait sera interpreté différemment par le logiciel. Mais ces deux dispositions donnent le même code en HTML qui est le suivant :

```
<td>cellule1</td><td>cellule2</td><td>cellule3</td>
```

Comme pour le tableau en lui-même, chacune des cellules d'un tableau peut contenir des paramètres. Pour cela, le contenu d'une cellule sera précédé des paramètres de la façon suivante :

```
|paramètres|cellule1
|paramètres|cellule2
|paramètres|cellule3
```

ou encore, suivant l'autre disposition :

```
|paramètres|cellule1||paramètres|cellule2||paramètres|cellule3
```

mais quelle que soit la disposition utilisée, le code HTML sera le suivant :

```
<td paramètres>cellule1</td><td paramètres>cellule2</td><td
paramètres>cellule3</td>
```

tr

Ce sont les balises `tr` qui permettent de passer à une nouvelle ligne dans le tableau. Leur utilisation est très simple, lorsque vous souhaitez passer à une nouvelle ligne, il suffit de faire :

```
|-
```

et comme le nombre de tiret - n'a pas de limite maximale, on peut très bien utiliser :

```
|----------------------------------------------------
```

et nous obtenons en HTML :

```
<tr> ... </tr>
```

Un grand avantage du codage wiki est qu'il se charge entièrement de la fermeture des balises `tr` ainsi que de leur ouverture si cette ouverture est nécessaire. Par exemple, si on reprend le tout premier tableau que nous avons créé, celui contenant une seule cellule, nous n'utilisons pas |- dans notre encodage et pourtant le code source HTML contient bien une balise `tr`.

Comme pour les autres balises, il est bien sûr possible d'utiliser des paramètres :

```
|- paramètres
```

ce qui donne en HTML :

```
<tr paramètres> ... </tr>
```

th

Les balises `th` sont très semblables aux balises `td` dans le sens où leur utilisation est totalement identique, seul la syntaxe change. Cependant il y a bien une différence notable entre ces deux balises qui intervient au niveau visuel. La balise `th` est généralement utilisée pour spécifier qu'une cellule représente un titre. Et elle peut, à cet effet, être utilisée partout où vous estimez que la cellule représente un titre. Visuellement, cela se voit car le texte de la cellule est en gras.

La syntaxe est différente et là où on utilisait un trait vertical pour les cellules, nous utiliserons un point d'exclamation !

```
!titre1
!titre2
!titre3
```

ou suivant l'autre disposition :

```
!titre1!!titre2!!titre3
```

par contre, entre les paramètres et le contenu de la cellule, nous utiliserons toujours le trait vertical |

```
!paramètres|titre1!!paramètres|titre2!!paramètres|titre3
```

et nous obtenons alors en HTML :

```
<th paramètres>titre1</th><th paramètres>titre2</th><th
paramètres>titre3</th>
```

Voyons ce que nous obtenons sur un exemple. Si on souhaite que dans notre tableau, la première ligne soit interpretée comme étant une ligne de titres :

Résultat affiché	Codage Wiki	Codage HTML

Titre Colonne 1 / **Titre Colonne 2**
cellule1 / cellule2
	`{| border="1"`
	`! Titre Colonne 1 !! Titre Colonne 2`
	`|-`
	`|cellule1`
	`|cellule2`
	`|}`
		`<table border="1">`
		`  <tr>`
		`  <th>Titre Colonne 1</th>`
		`  <th>Titre Colonne 2</th>`
		`  </tr>`
		`  <tr>`
		`  <td>cellule1</td>`
		`  <td>cellule2</td>`
		`  </tr>`
		`</table>`

Dans l'exemple qui suit, bien que la cellule de la seconde colonne ne soit pas balisée comme un titre (nous utilisons les deux traits || au lieu de !!), le logiciel la considère malgré tout comme telle :

Résultat affiché	Codage Wiki	Codage HTML

Titre Colonne 1 / **Colonne 2**	`{| border="1"`
	`! Titre Colonne 1 || Colonne 2`
	`|}`
		`<table border="1">`
		`  <tr>`
		`  <th>Titre Colonne 1</th>`
		`  <th>Colonne 2</th>`
		`  </tr>`
		`</table>`

Pour éviter ce problème, nous devons faire passer la deuxième colonne à la ligne comme ceci :

Résultat affiché	Codage Wiki	Codage HTML

Titre Colonne 1 / Colonne 2	`{| border="1"`
	`! Titre Colonne 1`
	`| Colonne 2`
	`|}`
		`<table border="1">`
		`  <tr>`
		`  <th>Titre Colonne 1</th>`
		`  <td>Colonne 2</td>`
		`  </tr>`
		`</table>`

caption

L'élément `caption` représente une légende. C'est un petit texte qui est placé au-dessus du tableau qui permet de préciser ce qui se trouve dans le tableau. Voici comment introduire une légende en Wiki :

`|+ légende`

qui donnera en HTML

`<caption>légende</caption>`

encore une fois, il est possible d'utiliser des paramètres de la façon suivante :

`|+ paramètres|légende`

qui donne alors

```
<caption paramètres>légende</caption>
```

Il n'est autorisé de placer qu'une seule légende par tableau. Cependant s'il en existe plusieurs, seule la première sera prise en compte. Voici le résultat que l'on obtient :

Résultat affiché	Codage Wiki	Codage HTML

légende
Colonne 1 | Colonne 2
	`{| border="1"`
	`|+ légende`
	`| Colonne 1`
	`| Colonne 2`
	`|}`
		`<table border="1">`
		`<caption>légende</caption>`
		`  <tr>`
		`  <td>Colonne 1</td>`
		`  <td>Colonne 2</td>`
		`  </tr>`
		`</table>`

Exemple récapitulatif

Maintenant que nous avons vu les différentes balises qui sont supportées par le codage Wiki, nous allons vous montrer un exemple récapitulatif dans lequel tout ce que nous venons de voir apparaîtra.

Résultat affiché	Codage Wiki	Codage HTML

légende
| **Titre 1** | **Titre 2** |
| **Titre ligne** | Cellule 1 | Cellule 2 |
	`{| border="1"`
	`|+ légende`
	`!`
	`! Titre 1 !! Titre 2`
	`|-`
	`! Titre ligne`
	`| Cellule 1`
	`| Cellule 2`
	`|}`
		`<table border="1">`
		`<caption>légende</caption>`
		`  <tr>`
		`  <th></th>`
		`  <th>Titre 1</th>`
		`  <th>Titre 2</th>`
		`  </tr>`
		`  <tr>`
		`  <th>Titre ligne</th>`
		`  <td>Cellule 1</td>`
		`  <td>Cellule 2</td>`
		`  </tr>`
		`</table>`

Tableaux imbriqués

Lorsque vous créez un tableau, vous pouvez y imbriquer un autre tableau. Au lieu de placer du texte dans une cellule, il suffit d'y placer un tableau comme le montre l'exemple suivant :

Résultat affiché	Codage Wiki	Codage HTML

<table>
<tr><td>Gauche</td><td><table><tr><td>tableau imbriqué</td><td>autre cellule</td></tr></table></td><td>Droite</td></tr>
<tr><td>Cellule 1</td><td>Cellule 2</td><td>Cellule 3</td></tr>
</table>

```
{| border="1"
| Gauche
|
{| border="1"
|tableau imbriqué
|autre cellule
|}
| Droite
|-
| Cellule 1
| Cellule 2
| Cellule 3
|}
```

```
<table border="1">
  <tr>
    <td>Gauche</td>
    <td>
      <table border="1">
        <tr>
          <td>tableau imbriqué</td>
          <td>autre cellule</td>
        </tr>
      </table>
    </td>
    <td>Droite</td>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
    <td>Cellule 3</td>
  </tr>
</table>
```

Les paramètres

Jusqu'à présent, nous avons vu la syntaxe pour créer des tableaux. Pour chacune des balises, nous avons précisé que l'on pouvait ajouter des paramètres. Ces différents paramètres seront étudiés dans cette section. Ils permettent notamment de spécifier le positionnement du tableau (gauche, milieu, droit), des couleurs à utiliser et bien d'autres choses encore.

border =

Ce paramètre que nous avons déjà utilisé dans nos différents tableaux permet de spécifier la taille de la bordure du tableau. Nous pouvons décider de ne pas placer de bordure en mettant une taille de 0.

`border="x"`

où x est donc la taille de la bordure.

Résultat affiché	Codage Wiki	Codage HTML

sans bordure 1 sans bordure 2	`{| border="0"`
	`|sans bordure 1`
	`|sans bordure 2`
	`|}`
		`<table border="0">`
		`  <tr>`
		`  <td>sans bordure 1</td>`
		`  <td>sans bordure 2</td>`
		`  </tr>`
		`</table>`

bordure normale 1 | bordure normale 2	{| border="1" |bordure normale 1 |bordure normale 2 |}	<table border="1">
<tr>
<td>bordure normale 1</td>
<td>bordure normale 2</td>
</tr>
</table>
grosse bordure 1 | grosse bordure 2	{| border="10" |grosse bordure 1 |grosse bordure 2 |}	<table border="10">
<tr>
<td>grosse bordure 1</td>
<td>grosse bordure 2</td>
</tr>
</table>

rowspan et colspan

Ces deux paramètres permettent de fusionner des cellules. Ils s'utilisent de la façon suivante :

```
colspan="x"
rowspan="x"
```

où x représente le nombre de cellules fusionnées. Voyons cela sur un exemple dans lequel on va fusionner deux colonnes :

Résultat affiché	Codage Wiki	Codage HTML
Simple | Fusionnée cellule1 | cellule2 | cellule3	{| border="1" | Simple | colspan="2" | Fusionnée |- | cellule 1 | cellule 2 | cellule 3 |}	<table border="1">
<tr>
<td>Simple</td>
<td colspan="2">Fusionnée</td>
</tr>
<tr>
<td>Cellule 1</td>
<td>Cellule 2</td>
<td>Cellule 3</td>
</tr>
</table>

Et voyons maintenant comment fusionner des lignes :

Résultat affiché	Codage Wiki	Codage HTML

Première Deuxième Gauche 1 Gauche 2 Gauche 3 Droite	{| border="1" | Première | Deuxième |- | Gauche 1 | rowspan="3" | Droite |- | Gauche 2 |- | Gauche 3 |}	<table border="1"> <tr> <td>Première</td> <td>Deuxième</td> </tr> <tr> <td>Gauche 1</td> <td rowspan="3">Droite</td> </tr> <tr> <td>Gauche 2</td> </tr> <tr> <td>Gauche 3</td> </tr> </table>

On voit donc avec cet exemple que l'on peut aisément fusionner des cellules que ce soit au niveau des colonnes ou au niveau des lignes. Le nombre de cellules fusionnées n'est pas limité. Et enfin, il est tout à fait possible de mélanger ces deux paramètres.

align et valign

Jusqu'à présent, tous nos tableaux étaient positionnés à gauche. Et le contenu de chacune des cellules était également positionné à gauche. Nous allons maintenant voir comment positionner aussi bien le tableau que le contenu des cellules. Un tableau ne peut-être aligné que suivant l'axe horizontal, c'est-à-dire à gauche, à droite ou au milieu. Alors que pour le contenu d'une cellule, on peut aussi décider de le positionner en haut, en bas ou au milieu de la cellule.

Pour l'alignement horizontal, il faut utiliser le paramètre `align` alors que pour l'alignement vertical, il faut utiliser `valign` :

```
align="left"
align="center"
align="right"
```

```
valign="top"
valign="center"
valign="bottom"
```

Voyons maintenant un exemple de positionnement d'un tableau :

Résultat affiché	Codage Wiki	Codage HTML
à gauche	{| align="left" border="1" |à gauche |}	<table align="left" border="1"> <tr> <td>à gauche</td> </tr> </table>

au centre	{| align="center" border="1" |au centre |}	<table align="center" border="1"> <tr> <td>au centre</td> </tr> </table>
à droite	{| align="right" border="1" |à droite |}	<table align="right" border="1"> <tr> <td>à droite</td> </tr> </table>

Passons au positionnement horizontal du texte dans une cellule. Les trois positions que nous montrons dans l'exemple qui suit sont donc gauche, milieu, droite.

Résultat affiché	Codage Wiki	Codage HTML
Positionnement g c d	{| border="1" | Positionnement |- | align="left" | g |- | align="center" | c |- | align="right" | d |}	<table border="1"> <tr> <td>Positionnement</td> </tr> <tr> <td align="left">g</td> </tr> <tr> <td align="center">c</td> </tr> <tr> <td align="right">r</td> </tr> </table>

Pour chacune des cellules, il est également possible de positionner le contenu verticalement. Dans l'exemple qui suit, nous avons créé une cellule (celle de gauche) avec une taille de 150 pixels pour que l'on puisse bien voir comment se positionne le texte des autres cellules.

Résultat affiché	Codage Wiki	Codage HTML

Position	haut	milieu	bas

```
{| border="1"
| height="150" | Position
| valign="top" | haut
| valign="center" |
milieu
| valign="bottom" | bas
|}
```

```
<table border="1">
  <tr>
    <td
height="150">Position</td
>
    <td
valign="top">haut</td>
    <td
valign="center">milieu</t
d>
    <td
valign="bottom">bas</td>
  </tr>
</table>
```

cellspacing et cellpadding

Il est possible de changer l'espacement entre les cellules grâce au paramètre `cellspacing`. Et il est possible de spécifier l'espacement entre le bord d'une cellule et son contenu grâce au paramètre `cellpadding`. Leur utilisation est la suivante :

`cellspacing="x"`

`cellpadding="x"`

Commençons par un exemple avec le paramètre `cellspacing` :

Résultat affiché	Codage Wiki	Codage HTML

cellule 1 cellule 2
cellule 3 cellule 4
	`{| border="1"`
	`cellspacing="5"`
	`|cellule 1`
	`|cellule 2`
	`|-`
	`|cellule 3`
	`|cellule 4`
	`|}`
		`<table border="1"`
		`cellspacing="5">`
		`  <tr>`
		`  <td>cellule 1</td>`
		`  <td>cellule 2</td>`
		`  </tr>`
		`  <tr>`
		`  <td>cellule 3</td>`
		`  <td>cellule 4</td>`
		`  </tr>`
		`</table>`

cellule 1 cellule 2
cellule 3 cellule 4
	`{| border="1"`
	`cellspacing="20"`
	`|cellule 1`
	`|cellule 2`
	`|-`
	`|cellule 3`
	`|cellule 4`
	`|}`
		`<table border="1"`
		`cellspacing="20">`
		`  <tr>`
		`  <td>cellule 1</td>`
		`  <td>cellule 2</td>`
		`  </tr>`
		`  <tr>`
		`  <td>cellule 3</td>`
		`  <td>cellule 4</td>`
		`  </tr>`
		`</table>`

et alors le paramètre `cellpadding` :

Résultat affiché	Codage Wiki	Codage HTML

cellule 1 | cellule 2
cellule 3 | cellule 4
	`{| border="1" cellpadding="5"`
	`|cellule 1`
	`|cellule 2`
	`|-`
	`|cellule 3`
	`|cellule 4`
	`|}`
		`<table border="1" cellpadding="5">`
		`  <tr>`
		`  <td>cellule 1</td>`
		`  <td>cellule 2</td>`
		`  </tr>`
		`  <tr>`
		`  <td>cellule 3</td>`
		`  <td>cellule 4</td>`
		`  </tr>`
		`</table>`

cellule 1 | cellule 2
cellule 3 | cellule 4
	`{| border="1" cellpadding="20"`
	`|cellule 1`
	`|cellule 2`
	`|-`
	`|cellule 3`
	`|cellule 4`
	`|}`
		`<table border="1" cellpadding="20">`
		`  <tr>`
		`  <td>cellule 1</td>`
		`  <td>cellule 2</td>`
		`  </tr>`
		`  <tr>`
		`  <td>cellule 3</td>`
		`  <td>cellule 4</td>`
		`  </tr>`
		`</table>`

width et height

Grâce aux paramètres `width` et `height`, on peut spécifier la largeur et la hauteur aussi bien du tableau en lui même que pour chacune des cellules du tableau. Lorsque l'on précise la taille, on peut préciser une valeur absolue en pixels ou bien on peut également préciser une taille en pourcentage. Dans les deux cas, cela représente une taille minimale, cependant si le contenu du tableau est trop gros, ce tableau adaptera sa taille au contenu.

Le premier exemple montre l'utilisation d'une taille fixe. Le premier cas utilise une largeur de 10 pixels, cependant la largeur du contenu du tableau étant plus grande, il s'adapte donc à cette taille. Le second cas utilise une taille de 100 pixels ; ici nous n'avons pas de problème car 100 pixels est bien plus grand que la largeur occupée par le contenu du tableau, celui-ci a donc bien une largeur de 100 pixels. Et enfin le troisième exemple utilise une largeur de 200 pixels.

Résultat affiché	Codage Wiki	Codage HTML

cellule	`{| border="1" width="10"`
	`|cellule`
	`|}`
		`<table border="1" width="10">`
		`  <tr>`
		`  <td>cellule</td>`
		`  </tr>`
		`</table>`

cellule	`{| border="1" width="100"`
	`|cellule`
	`|}`
		`<table border="1" width="100">`
		`  <tr>`
		`  <td>cellule</td>`
		`  </tr>`
		`</table>`

cellule	{| border="1" width="200" |cellule |}	<table border="1" width="200"> <tr> <td>cellule</td> </tr> </table>

Voyons maintenant l'utilisation d'une taille proportionnelle. Dans le premier cas, on impose au tableau d'occuper un maximum de 33% de la largeur disponible. Dans le deuxième, puis le troisième cas, le tableau doit respectivement occuper un maximum de 50%, puis de 100%, de l'espace disponible.

Résultat affiché	Codage Wiki	Codage HTML
cellule	{| border="1" width="33%" |cellule |}	<table border="1" width="33%"> <tr> <td>cellule</td> </tr> </table>
cellule	{| border="1" width="50%" |cellule |}	<table border="1" width="50%"> <tr> <td>cellule</td> </tr> </table>
cellule	{| border="1" width="100%" |cellule |}	<table border="1" width="100%"> <tr> <td>cellule</td> </tr> </table>

Pour le paramètre `height`, l'utilisation est la même que pour `width` mais cette fois-ci on changera alors la hauteur du tableau. Ces deux paramètres peuvent également être utilisés pour chacune des cellules. Voyons un exemple avec les cellules :

Résultat affiché	Codage Wiki	Codage HTML

Tableau 1 1/3 cellule 1 | 1/3 cellule 2 | 1/3 cellule 3	{| border="1" width="100%" | colspan="3" | Tableau 1 |- | width="33%" | 1/3 cellule 1 | width="33%" | 1/3 cellule 2 | width="33%" | 1/3 cellule 3 |}	<table border="1" width="100%"> <tr> <td colspan="3">Tableau 1</td> </tr> <tr> <td width="33%">1/3 cellule 1</td> <td width="33%">1/3 cellule 2</td> <td width="33%">1/3 cellule 3</td> </tr> </table>
Tableau 2 1/2 cellule 1 | 1/4 cellule 2 | 1/4 cellule 3	{| border="1" width="100%" | colspan="3" | Tableau 2 |- | width="50%" | 1/2 cellule 1 | width="25%" | 1/4 cellule 2 | width="25%" | 1/4 cellule 3 |}	<table border="1" width="100%"> <tr> <td colspan="3">Tableau 2</td> </tr> <tr> <td width="50%">1/2 cellule 1</td> <td width="25%">1/4 cellule 2</td> <td width="25%">1/4 cellule 3</td> </tr> </table>

bgcolor

Il est possible de changer la couleur de fond d'une cellule. Pour cela, il faut utiliser le paramètre `bgcolor` de la façon suivante :

`bgcolor="#hex"`

Pour spécifier une couleur, il faut donc utiliser le caractère # suivi du code couleur en hexadécimal. Reportez-vous à la page Aide:couleurs pour avoir une liste de couleurs avec leur valeur hexadécimale. Pour bien illustrer cela, regardons l'exemple qui suit :

Résultat affiché	Codage Wiki	Codage HTML

rouge vert bleu	{| border="1" | bgcolor="#FF0000" | rouge | bgcolor="#00FF00" | vert | bgcolor="#0000FF" | bleu |}	<table border="1"> <tr> <td bgcolor="#FF0000">rouge</td> <td bgcolor="#00FF00">vert</td> <td bgcolor="#0000FF">bleu</td> </tr> </table>

style

Nous arrivons au dernier des paramètres que l'on peut utiliser dans des tableaux : `style`. Ce paramètre permet de changer tout l'aspect graphique. Il peut aussi bien être utilisé pour le tableau que pour une cellule unique. Voici un exemple d'utilisation du paramètre style, nous allons changer la couleur de fond du tableau et choisir la couleur verte :

Résultat affiché	Codage Wiki	Codage HTML
cellule 1 cellule 2 cellule 3	{| border="1" style="background-color:#CCFFCC" | cellule 1 |- | cellule 2 |- | cellule 3 |}	<table border="1" style="background-color:#CCFFCC"> <tr> <td>cellule 1</td> </tr> <tr> <td>cellule 2</td> </tr> <tr> <td>cellule 3</td> </tr> </table>

Remarques

Quand les tableaux sont-ils appropriés ?

Les tableaux sont parfaits pour organiser toute information qui sera plus clairement présentée dans un format lignes/colonnes. Ce qui inclut :

- Tableaux mathématiques
  - Tables de multiplication
  - Tables de division
  - Tableaux de recherche
- Listes d'informations
  - Mots équivalents dans deux ou plusieurs langues
  - Personnalités, date de naissance, fonction
  - Artiste, disque, année et maison d'édition

Très souvent, une liste est plus lisible laissée en tant que liste. Certains articles comportent des listes excessivement longues très difficiles à éditer si elles se retrouvent sous forme de tableaux. Avant de transformer une liste en tableau, essayez de vous figurer l'aspect sous forme de tableau

(lignes et colonnes) afin d'être sûr de son utilité. Si c'est le cas, alors l'option du tableau est certainement le meilleur choix.

Quand les tableaux sont-ils inappropriés ?

Les tableaux ne doivent pas être utilisés uniquement pour la présentation. Si l'information que vous éditez n'est pas de nature tabulaire, elle ne devra alors pas être mise en tableau. Évitez d'utiliser les tableaux pour mettre une légende sous une illustration, ordonner un groupe de liens ou autres cas strictement visuels. Cela rend l'article trop difficile à éditer pour d'autres Wikipédiens et les tableaux ne sont pas vraiment faits pour cela.

Liste très longues ou listes très courtes

Si une liste est vraiment très longue ou au contraire très simple, préférez l'usage des formats de listes standards de Wikipédia. Les listes longues seront difficiles à maintenir si elles se retrouvent à l'intérieur de tableaux et les listes courtes sont réellement trop simples pour nécessiter un formatage en tableau. Voici un petit exemple à ne pas suivre :

1980	Vague de pluie
1988	Nom d'un chat !
1994	La vie, la pomme
1994	Le ciel est vert

choisissez plutôt une liste classique :

- 1980 : Vague de pluie
- 1988 : Nom d'un chat !
- 1994 : La vie, la pomme
- 1994 : Le ciel est vert

Légende d'illustration

Souvent, les illustrations d'un article sont placées dans un semblant de tableau. Du fait que le tableau peut être flottant et placé à gauche ou à droite de l'écran, il paraît facile et pratique d'utiliser un tableau unicellulaire pour placer l'image à un endroit précis de l'écran. Cette solution était nécessaire pour les anciens navigateurs, particulièrement ceux qui ne supportaient pas les feuilles de style en cascade pour l'affichage de telles images. Aujourd'hui, la plupart des navigateurs fonctionnent parfaitement avec ces feuilles de style. Aussi, la façon recommandée d'afficher les images est d'utiliser les balises nommées `div`.

Voici un petit aperçu à ne pas faire :

```
{| align="right" border="0" cellpadding="0" | [[File:Blueberries.PNG]] |}
```

mais procédez de la façon suivante qui est tout à fait correcte :

```
[[File:Blueberries.PNG|right|]]
```

Dans ces deux cas, le résultat sera le même. L'illustration sera flottante à droite de l'écran et le texte environnant habillera l'illustration. Voici ce que cela donne dans votre navigateur (avec du texte ajouté) :

La myrtille (du latin myrtus, du grec murtos : myrte, symbole de l'amour ou de la gloire) est l'espèce Vaccinium myrtillus, airelle également connue sous les dénominations de airelle myrtille, arbrêtier, gueule noire, mauret, brimbelle ou raisin des bois. Au Canada, on l'appelle bleuet et sa culture est très importante, notamment celle des fruits sauvages. Le nom myrtille désigne aussi bien le végétal que son fruit, mais le terme de myrtillier est cependant usité.

La myrtille est un vigoureux sous-arbrisseau vivace et rampant, de 20 à 60 cm, que l'on trouve en Eurasie et en Amérique du Nord et qui forme des fourrés nains en dressant des rameaux serrés aux tiges vertes à section triangulaire. En France, elle est commune en montagne, surtout dans la moitié Nord de la France. Elle croît entre 400 à 2500 m d'altitude dans les forêts de conifères, les bois clairs, les landes et les tourbières, associée aux plantes acidophiles.

Dans le langage des fleurs, la myrtille signifie que l'on recherche la solitude.

Problèmes éventuels

Les tableaux peuvent provoquer d'autres difficultés, surtout lorsqu'ils ne sont pas correctement utilisés. Voici quelques cas que vous pourrez rencontrer lors de l'usage de tableaux dans vos articles :

- Les tableaux peuvent être une difficulté pour certains éditeurs, spécialement pour les nouveaux Wikipédiens. Les nouveaux éditeurs peuvent être effrayés lorsqu'ils cliquent sur

« Modifier cette page » et qu'ils découvrent un gros pavé inintelligible (pour eux) de codes HTML. Essayez de conserver des tableaux simples et respectant la codification. Vous pouvez également ajouter des commentaires (qui n'apparaîtront pas dans la page normale) du genre « <!-- Pour éditer le texte de cet article, dépassez le tableau. --> » afin de rassurer les éditeurs.

- Il est souvent difficile, même pour des auteurs expérimentés en HTML, d'être sûr de l'aspect qu'aura le tableau dans tous (ou beaucoup) de navigateurs. Même la plus petite erreur typographique peut avoir des conséquences catastrophiques sur l'affichage du tableau. Même si vous êtes confiant dans votre aptitude à éviter ce genre de problème, il se peut que ce ne soit pas le cas de futurs éditeurs. Encore une fois, essayez d'élaborer des tableaux simples et bien écrit afin de minimiser les risques.
- Les gros tableaux contenant beaucoup d'informations peuvent dépasser le bord droit de l'écran sur de basses résolutions d'affichage. Cela peut parfois être acceptable, spécialement si le lecteur y a été préparé (notamment lorsque l'on sait par avance qu'un tableau, comme la Table des isotopes (complète), est volontairement très grand). Si vous devez, pour votre article, créer un tableau vraiment très grand, vous devrez alors considérer qu'il y aura lieu d'en créer un autre plus simple, plus petit, pour les utilisateurs qui ne pourront pas visualiser la version longue.
- Si vous insérez du texte à chasse fixe dans un tableau (en utilisant les balises HTML `code`, `pre` ou `tt`, par exemple), cela forcera la page à être plus large que nécessaire. Tant que cela est possible, évitez l'utilisation de tels textes dans vos tableaux, ainsi le texte sera adapté naturellement. Un problème identique survient lors de l'insertion d'images dans un tableau (car les images sont généralement contraintes à une taille fixe).
- Des cellules contenant une grosse quantité d'informations peuvent causer des problèmes d'affichage dans certains navigateurs. En particulier, des cellules contenant un grand paragraphe risquent de brouiller l'affichage de navigateurs en mode console comme Lynx ou Links. De manière générale, tant que faire se peut, tentez de limiter la quantité de texte à l'intérieur d'un tableau.

synthèse / aide-mémoire

fonction	code	note
	s t r u c t u re	

délimitation du tableau	`{|`
	`tableau`
	`|}`	

tableau paramétré	`{| params`
	`tableau`
	`|}`	

légende (caption)	`|- légende`	
légende paramétrée	`|- params | légende`	à placer en début de ligne
balise de nouvelle ligne	`|-`	à placer en début de ligne ; sans cellule
nouvelle ligne paramétrée	`|- params`	
cellule titre	`!titre`	style gras & centré par défaut
cellule titre paramétrée	`!params|titre`	
ligne titres (en-tête)	`!titre1!!titre2!!titre3`	

cellule	`|cell`	
cellule paramétrée	`|params|cell`	
ligne	`!cell1||cell2||cell3`	
f o r m a t		
largeur globale	`width="n%"`	fonction de la fenêtre du navigateur
cadre & grille	`border="n"`	épaisseur seulement pour cadre
cadre seulement	`style="border:n solid couleur"`	couleur hexa (cf + bas second tableau)
écart entre cellules	`cellspacing="n"`	cadre de cellule si n>0 (cf ci-dessous)
cadre seulement (alternative)	`border="n" cellspacing="0"`	demande écart nul entre cellules
tampon intérieur cellule	`cellpadding="n"`	entre cadre & texte
étendue de cellule H ou V	`colspan="n" rowspan="n"`	lire : cellule couvre n colonnes ou n lignes !!!
flottement & positionnement gauche/droite du tableau	`align="côté"`	le texte extérieur au tableau passe sur le côté
centrage du tableau	`style="margin: 1em auto 1em auto"`	c'est une astuce --- le texte extérieur ne passe pas sur les côtés
titre du tableau (légende en haut)	+titre	titre collé-centré-gras
en-têtes	`!en-tête`	pour colonne & ligne
alignement vertical ligne ou cellule	`valign="top"`	*pas pour tableau entier (?)*
s t y l e		
format global params de style	**`style = "param:val; param:val; ...;"`**	**pour tableau, ligne, cellule attention : points-virgules collés !!!**
hauteur & largeur cellules	`style="height:100px; width:100px;"`	on peut aussi mesurer en %age
alignement du texte	`style="text-align:center"`	
code couleur	6 chiffres hexa :#rrvvbb	ex : #789abc
couleur de fond	`style="background:couleur"`	
couleur de texte	`style="color:couleur"`	

style texte	`style="font-style:italic; font-size:150%;"`	
alignement texte (alternative)	`align="left"`	
couleur de fond (alternative)	`bgcolor="couleur"`	
couleur de texte (alternative)	`<span style="color:couleur">texte</span>`	codage (x)html

Commentaire

Il est possible d'ajouter des commentaires dans la page, qui n'apparaîtront pas dans le rendu de la page, mais seulement en mode édition.

```
<!-- Note : Cette page est en cours de rédaction, veuillez patienter avant de l'éditer. -->
```

Catégories

Pas besoin de coder les catégories, il suffit de la/les sélectionner dans la liste déroulante.

WIKI - KMS - Crontabs

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Linux
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	Documentation
Status:	Active
Access:	free

Download this page as PDF

== Introduction == The following scripts need to be scheduled on the server to maintain data consistencies :

- RunJobs.sh
- BuildSemanticData.sh
- Indexation.sh

These are PHP scripts but to ease the work of non-mediawiki addict admins, they are summarizede

in these sh scripts.

RunJobs.sh

Even if a service is running in the background, it is necessary to run periodically because it is required for Semantic Data to be correct.

Code script RunJobs.sh

```
cd
clear
cd /usr/local/apache2/htdocs/mediawiki_latest/maintenance/
php runJobs.php
cd
```

cd clear cd /usr/local/apache2/htdocs/mediawiki_latest/maintenance/ php runJobs.php cd

Indexation.sh

Indexation.sh needs to be run in order to rebuild Mediawiki's indexes. Because the database is on an instable cluster, they often get corrupted (around once or twice a week in general).

Script Indexation.sh

```
cd
clear
cd /usr/local/apache2/htdocs/mediawiki_latest/maintenance/
php rebuildall.php
cd
```

BuildSemanticData.sh

BuildSemanticData.sh suffers not only from instable connections to the database cluster, but is also dependent on the cache of the browser of each user. Therefore, datas may look incomplete or different according who is watching what. To resolve this, the BuildSemanticData.sh scrip^t need to be ran regularly (2 or 3 times a day right after the RunJobs.sh script.

Code script BuildSemanticData.sh

```
cd
clear
cd
/usr/local/apache2/htdocs/mediawiki_latest/extensions/SemanticMediaWiki/maint
enance/
php rebuildData.php
cd
```

Crontab script

One script, ScriptingBulk.sh, is to be scheduled in root on each server (DEV, ACC, PRD) to run daily at It starts each of the 3 preceeding script in the following oprder :

1. RunJobs.sh
2. Indexation.sh
3. BuildSemantiocData.sh

Code script ScriptingBulk.sh

```
cd
clear
cd /usr/local/apache2/htdocs/scripting
sudo ./RunJobs.sh
sudo ./Indexation.sh
sudo ./BuildSemanticData.sh
cd
```

Scheduling the scripts to run

The contab created as root (sudo su -)is the following :

```
30 5,12,17 * * * /usr/local/apache2/htdocs/scripting/ScriptingBulk.sh
```

This means that the Mediawiki php scripts will run daily at 05:30, 12:30 and 17:30.

cron is a Unix daemon that allows users to schedule programs to run at set intervals. A **crontab** is the file that cron uses to determine what actions to perform and when. A cronjob is a the specific line (or job) in the crontab. The Toolserver provides a version of cron called **cronie**, which is more featureful than Solaris' built-in cron, and will be described here. For information on Solaris cron, refer to the **crontab(1)** manual page.

On the Toolserver, we **strongly** recommend that cron be used in conjunction with job scheduling. This is how to do this.

Basic commands

crontab has three basic commands:

`$ crontab -e`
: edits (or creates) your current crontab in a text editor

`$ crontab -l`
: displays your crontab

`$ crontab -r`
: removes your crontab entirely

You can change the text editor used for editing the crontab in your environment.

crontab syntax

Each line in the crontab file is a single cron job. The line contains five fields at the start indicating when the job should run, and the rest of the line is the command.

```
minute (0-59),
|       hour (0-23),
|       |       day of the month (1-31),
|       |       |       month of the year (1-12),
|       |       |       |       day of the week (0-6 with 0=Sunday).
|       |       |       |       |       commands
30      2       *       *       0,6     /some/command/to/run
30      2       *       *       1-5     /another/command/to/run >/dev/null
```

Use "*" to mean "every"; for example, "30 2 * * *" runs the program at 02:30 UTC, every day of every month. Only specify "day of month" *or* "day of week", not both.

On Linux, you can write "*/**x**" to mean "every **x**". For example, writing "*/10" in the "minute" column means "every ten minutes".

The crontab file must end with a blank line (that is, the last character of the final line in the file must be a newline character). Many editors, like **nano** (the default editor) and **vi** do this by default. The "**joe**" editor does *not* do this - be sure to end your file with a blank line every time. If you don't, you won't receive an error message, but nothing will happen at the scheduled time.

An example which would run every hour of the day would be:

```
0 * * * * php $HOME/scripts/myscript.php
```

This means every hour, i.e. *, at minute 0. There is no need to write */1 for the hour field, as */1 is identical to *.

Lines beginning with a # will be treated as comments, and ignored, along with blank lines.

Working with Mediawiki

Mediawiki Extension

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Book
Last Edit By:	DochyJP
LastEdit:	2021-02-26
Document type:	User Guide
Status:	Active
Access:	Micylou-restricted

Download this page as PDF

load PDF

Back to top of page - Back to Welcome Page