https://wiki.micylou.com/index.php/WIKI_-_KMS_-_Crontabs

WIKI - KMS - Crontabs

WIKI - KMS - Crontabs

Source:	DataSource
Language:	English
Topic:	Mediawiki
SubTopic:	Linux
Last Edit By:	DochyJP
LastEdit:	2021-04-14
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.

□

Contents

- 1 RunJobs.sh
  - 1.1 Code script RunJobs.sh
- 2 Indexation.sh
  - 2.1 Script Indexation.sh
- 3 BuildSemanticData.sh
  - 3.1 Code script BuildSemanticData.sh
- 4 Crontab script
  - 4.1 Code script ScriptingBulk.sh
- 5 Scheduling the scripts to run
- 6 Basic commands
- 7 crontab syntax

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.