

# Archive Manager

## What is it ?

Archive Manager is a tool developed in Python by [Maxime Perrin](#) to automatically manage the archiving of a compressed folder containing a SQL file.

## How to use it ?

- **Installation**

To install the utility on your computer or server, simply clone the git repository or download the project files.

```
git clone git@github.com:maximeperrindev/scripting-system.git
```

Once the project is imported, you have to configure the conf.ini file (an example file is provided in the repository "conf\_example.ini").

conf\_example.ini (don't forget to rename it conf.ini) :

```
[FILE]
URL = URL_TO_FILE
SQLFILE = NAME_OF_DUMP_FILE_WITH_EXTENSION
[SERVER]
URL = URL_TO_WEBDAV_SERVER_WITH_/webdav
USERNAME = USERNAME_OF_WEBDAV_SESSION
PASSWORD = USERNAME_OF_WEBDAV_SESSION
CONSERVATION_TIME = NUMBER_OF_DAYS
[MAIL]
ENABLED = TRUE
USERNAME = USERNAME(MAIL ADDRESS)
PASSWORD = PASSWORDFORMAIL
ATTACH_LOG = TRUE
MAIL_LIST = MAIL LIST WITH SPACES
[MATTERMOST]
WEBHOOK = WEBHOOK_URL
```

The most important point is to fill correctly this file. If it's not the case, Archive Manager could return a lot of errors.

To install the dependencies, you will find the requirements.txt file. You must run this command :

```
pip install -r requirements.txt
```

If an error occurred, you may have not installed correctly all modules. Check if all dependencies are correctly installed.

The utility is now set up and ready to work.

- **Launch**

To launch Archive Manager, you have to install Python 3 => [tutorial here](#).

Locate your terminal command in the project folder

```
cd path/to/project
```

Now, you can run

```
python3 archive.py
```

It will create :

```
* A file archive.log wich contains all logs related to the execution
* A resources folder wich will contains the downloaded zip file with the extracted SQL file
```

### CRON USE

You can launch automatically Archive Manager with a Cron instruction.

To do so, you firstly need to get the absolute path of Python executable

```
which python3
```

After, you need to get the absolute path of archive.py script (go in the project folder)

```
pwd
```

You can now edit the Crontab with

```
crontab -e
```

For example, to launch Archive Manager every day at 16:00:

```
00 16 * * * /path/to/python /path/to/archive.py
```

- **### Logs** All actions, errors caused by the execution of the program are listed in a log file "archive.log".

You can find an example here :

```
2021-12-21 16:01:37,081 - INFO - Starting script...
2021-12-21 16:01:37,302 - INFO - test_export.sql.zip(2188 bytes) downloaded with success
2021-12-21 16:01:37,318 - INFO - test_export.sql extracted with success in /resources/_MACOSX/_test_export.sql
2021-12-21 16:01:37,333 - INFO - Connecting to remote server...
2021-12-21 16:01:37,355 - INFO - Connected !
2021-12-21 16:01:37,368 - INFO - Checking for expired files...
2021-12-21 16:01:37,387 - INFO - Folder is up to date !
2021-12-21 16:01:37,403 - INFO - 20212112.tgz file created in ./resources
2021-12-21 16:01:37,421 - INFO - Uploaded !
2021-12-21 16:01:38,752 - INFO - Mattermost webhook notification done ! (https://chat.telecomste.fr/hooks/13ta3nw3e787tnxxqkyxfh5xn
```

This allows to immediately target the cause of a bug.

## Any problem ?

If a problem occurred during the use of Archive Manager, or if you have a question about the project, please contact the developer at [contact@maximeperrin.fr](mailto:contact@maximeperrin.fr)