

# PowerCompost

---

## Documentation des scripts python

Auteur(s) : Florentin Delaine  
Status : En cours de rédaction

---

### Résumé

L'objectif de ce document est de fournir toutes les informations nécessaires afin de pouvoir utiliser les scripts et éventuellement les adapter à ses besoins. Comme il le sera précisé, certaines fonctionnalités ne sont pas optimisées, peuvent présenter des risques, etc, il convient donc de savoir un minimum ce que l'on fait.

## Table des matières

<b>1</b>	<b>Ce qui est nécessaire</b>	<b>3</b>
<b>2</b>	<b>informationsBDD.py</b>	<b>3</b>
<b>3</b>	<b>scriptDecoupe.py</b>	<b>3</b>
<b>4</b>	<b>scriptExtractionDonneesOld.py</b>	<b>3</b>
4.1	Fichiers sources . . . . .	3
4.2	script . . . . .	5
<b>5</b>	<b>scriptExtractionDonneesNI.py</b>	<b>5</b>
<b>6</b>	<b>scriptExportCSV.py</b>	<b>5</b>
<b>7</b>	<b>Remarques générales</b>	<b>5</b>
	<b>Références</b>	<b>5</b>
<b>A</b>	<b>Sauvegarder une base de données MySQL</b>	<b>6</b>

## 1 Ce qui est nécessaire

- Un système de gestion de base de données si vous souhaitez travailler en local, ici MySQL. Installation ici ou de préférence avec Homebrew pour les Mac users.
- Python 3 et les packages `sp` et `pymysql`. `sp` est fourni avec le code, il faut par contre installer `pymysql`. Google peut vous aider.

*Remarque* : si vous n'avez pas envie de toucher au shell, l'installation de PyCharm CE est une solution. C'est un IDE que je trouve bien fichu pour Python. Dans les préférences, il permet d'installer les packages que l'on veut pour n'importe quelle version de Python (moyennant l'installation de quelques packages comme `pip` mais ça vous sera précisé).

Il va de soi que pour les sections suivantes, votre serveur MySQL doit être actif...

## 2 informationsBDD.py

Ce script permet de préciser les informations de connexion à la base de données : l'adresse du serveur, vos identifiants et la base de données sur laquelle agir. Attention à ne pas envoyer ce fichier sur git.

Les scripts d'import vous demandent une confirmation de ces informations avant d'agir pour éviter de foutre le bazar et votre mot de passe. Si vous mettez le bazar, vous pouvez aller chercher le goudron et les plumes.

## 3 scriptDecoupe.py

Ce fichier fournit une fonction pour découper un fichier en plusieurs petits. Pour modifier la taille des fichiers de sortie, regardez du côté des paramètres de la boucle dans `decoupe()`. Ces fichiers sont placés dans un dossier à l'intérieur du dossier `Decoupe`, il est donc important de conserver ce dossier.

Une fonction est également fournie pour supprimer tous ces petits fichiers une fois qu'ils ne servent plus.

## 4 scriptExtractionDonneesOld.py

Script pour l'import de données datant de l'époque où LabView était utilisé pour l'acquisition.

### 4.1 Fichiers sources

Les fichiers doivent porter l'extension `.txt` et ne pas comporter d'espaces dans leur nom (ni d'autres caractères exotiques).

Si le fichier comporte un en-tête du type :

```
1 LabVIEW Measurement
2 Writer_Version 2
```

```

3 Reader_Version  2
4 Separator      Tab
5 Decimal_Separator  ,
6 Multi_Headings No
7 X_Columns      One
8 Time_Pref      Absolute
9 Operator       poste2
10 Date          2013/09/10
11 Time          17:17:30,7267637252807617188
12 ***End_of_Header***
13
14 Channels       8
15 Samples 1      1      1      1      1      1      1
16 Date          2013/09/10      2013/09/10      2013/09/10      2013/09/10
                2013/09/10      2013/09/10      2013/09/10      2013/09/10
17 Time          17:17:31,3097968101501464844
                17:17:31,3097968101501464844      17:17:31,3097968101501464844
                17:17:31,3097968101501464844      17:17:31,3097968101501464844
                17:17:31,3097968101501464844      17:17:31,3097968101501464844
                17:17:31,3097968101501464844
18 X_Dimension   Time      Time      Time      Time      Time      Time
    Time      Time
19 X0            0,0000000000000000E+0      0,0000000000000000E+0
                0,0000000000000000E+0      0,0000000000000000E+0
                0,0000000000000000E+0      0,0000000000000000E+0
                0,0000000000000000E+0      0,0000000000000000E+0
20 Delta_X 1,000000      1,000000      1,000000      1,000000
                1,000000      1,000000      1,000000      1,000000
21 ***End_of_Header***
22 X_Value  Untitled      Untitled 1      Untitled 2      Untitled 3
                Untitled 4      Untitled 5      Untitled 6      Untitled 7
                Comment
23 0,000000      21,456231      21,951068      21,556587
                22,167919      21,569221      20,844871      21,732911
                2293,680716

```

Remplacez le par un en-tête du type

```

1
2 21/10/13      TC0      TC1      TC2      TC3      TC4      TC5      TC6
                TC7
3 0,000000      21,456231      21,951068      21,556587
                22,167919      21,569221      20,844871      21,732911
                2293,680716

```

où les valeurs sont séparées par des tabulations et le séparateur décimal, une virgule.

Aussi la syntaxe d'une ligne doit être de la forme suivante :

```

1 0,000000      21,456231      21,951068      21,556587
                22,167919      21,569221      20,844871      21,732911

```

2293,680716

*Remarque* : Pour l’instant le script plante si vous n’avez pas cette forme (date + capteurs notés TCx, une valeur de temps et 8 valeurs pour les capteurs). On peut imaginer un code plus général en modifiant la grammaire du parseur.

## 4.2 script

Placez votre fichier dans le même répertoire que le script `scriptExtractionDonneesOld.py`, lancez le et suivez les instructions. On peut aussi ne pas faire la copie mais il faudra préciser le path relatif ou absolu comme demandé au début du script.

## 5 scriptExtractionDonneesNI.py

Placez votre fichier dans le même répertoire que le script `scriptExtractionDonneesNI.py`, lancez le et suivez les instructions. On peut aussi ne pas faire la copie mais il faudra préciser le path relatif ou absolu comme demandé au début du script.

## 6 scriptExportCSV.py

On lance `scriptExportCSV.py`. On demande les tables qu’on souhaite exporter, les écarts entre les tables s’il y en a plusieurs, le pas de temps et le script fait le reste.

*Rq* : si la table ne commence pas à zéro et que c’est la première table, la première ligne de la table est recopiée jusqu’à ce qu’on soit à une itération dans le script qui permet de reprendre l’export normalement. On peut imaginer une solution plus esthétique, c’est ce qui a été implémenté dans l’application en C++.

## 7 Remarques générales

Ces scripts python sont lents. Cela pourrait être amélioré en utilisant la librairie `numpy`. À bon entendeur.

## Références

## A Sauvegarder une base de données MySQL

Sous Mac OS X, ouvrez Terminal et entrez les lignes suivantes en modifiant `USER_NAME`, `DATABASE_NAME`, `FILE_PATH` et `FILE_NAME`.

```
1 mysql -u USER_NAME -p DATABASE_NAME < FILE_PATH/FILE_NAME.sql
```

Sous Linux (non testé mais ça doit être ça), ouvrez Terminal et entrez les lignes suivantes en modifiant `USER_NAME`, `DATABASE_NAME`, `FILE_PATH` et `FILE_NAME`.

```
1 mysql -u USER_NAME -p DATABASE_NAME < FILE_PATH/FILE_NAME.sql
```

Sous Windows, ouvrez `cmd.exe` et entrez les lignes suivantes en modifiant `USER_NAME`, `DATABASE_NAME`, `FILE_PATH` et `FILE_NAME`. Modifiez aussi le path pour le répertoire de MySQL suivant la version que vous utilisez.

```
1 mysql.exe -u USER_NAME -p DATABASE_NAME < FILE_PATH/FILE_NAME.sql
```

On vous demande votre mot de passe utilisateur, vous l'entrez. Tout s'importe ensuite et c'est généralement rapide.