

WEBDYN
flexitron group

APPLICATION NOTE



WebdynEasy W M-Bus 868MHz

Helper Scripts Application Note

Contents

1	Introduction	3
2	Prerequisite	3
3	json2bson.py	4
4	bson2json.py	4
5	data2csv.py	5
6	filters.py	5
7	whiteList.py	5
8	csv2whiteList.py	8

1 Introduction

This application note presents a series of tiny python scripts intended to help users that are not familiar with WebdynEasy original file formatting or those who don't have a dedicated portal that can facilitate WebdynEasy usage, or to introduce portal developers to the manipulation of WebdynEasy files.

Scripts are available for download from Webdyn Website:

<https://www.webdyn.com/download/WebDynEasyScripts.zip>

Also useful, the WebdynEasy UserManual available on our support page:

[English version](#)

[French version](#)

[Deutsch version](#)

[Spanish version](#)

2 Prerequisite

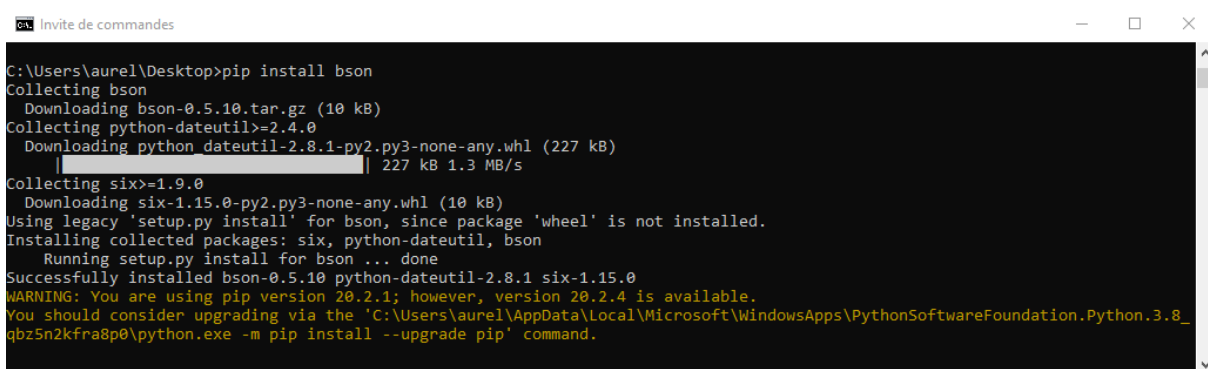
a) Windows

Install python version 3.8 or above:

<https://www.python.org/ftp/python/3.8.6/python-3.8.6-amd64.exe>

Then install following modules from command prompt:

- pip install bson
- pip install python-bsonjs



```
Invite de commandes
C:\Users\aurel\Desktop>pip install bson
Collecting bson
  Downloading bson-0.5.10.tar.gz (10 kB)
Collecting python-dateutil>=2.4.0
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
  | 227 kB 1.3 MB/s
Collecting six>=1.9.0
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Using legacy 'setup.py install' for bson, since package 'wheel' is not installed.
Installing collected packages: six, python-dateutil, bson
  Running setup.py install for bson ... done
Successfully installed bson-0.5.10 python-dateutil-2.8.1 six-1.15.0
WARNING: You are using pip version 20.2.1; however, version 20.2.4 is available.
You should consider upgrading via the 'C:\Users\aurel\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.8_
qbz5n2kfra8p0\python.exe -m pip install --upgrade pip' command.
```

```
C:\Users\auarel>pip install python-bsonjs
Collecting python-bsonjs
  Downloading python-bsonjs-0.2.0.tar.gz (150 kB)
    |██████████████████████████████| 150 kB 3.2 MB/s
Using legacy 'setup.py install' for python-bsonjs, since package 'wheel' is not installed.

Installing collected packages: python-bsonjs
  Running setup.py install for python-bsonjs ... done
Successfully installed python-bsonjs-0.2.0
WARNING: You are using pip version 20.2.1; however, version 20.2.4 is available.
You should consider upgrading via the 'C:\Users\auarel\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\python.exe -m pip install --upgrade pip'
command.
```

b) Linux

```
sudo apt install python3 sudo
apt install python3-pippip3
install bson
pip3 install python-bsonjs
```

3 [json2bson.py](#)

This script performs conversion from a json file to a bson file with crc

See script: <https://www.webdyn.com/download/json2bson.py>

```
usage:
    json2bson intpufilename.json outputfilename.bson

if the first arg is '-' stdin is used as input
if the second arg is '-' stdout is used as output
if the second arg is omitted the output file name will be: intpufilename.bson
```

4 [bson2json.py](#)

This script performs conversion from a bson file to a json file.

See script: <https://www.webdyn.com/download/bson2json.py>

```
usage:
    bson2json intpufilename.bson outputfilename.json

if the first arg is '-' stdin is used as input
if the second arg is '-' stdout is used as output
if the second arg is omitted the output file name will be: intpufilename.json
```

5 [data2csv.py](#)

This script performs conversion from a bson data file produced by the WebdynEasy to a csv file containing 1 telegram per line.

See script: <https://www.webdyn.com/download/data2csv.py>

```
usage:
    data2csv [-d] [-m] inputfile.data outputfile.csv

'-d' (optionnal) converts timestamp to a readable date
'-m' (optionnal) manufacturer code is converted to a readable value if
the first arg is '-' stdin is used as input
if the second arg is '-' stdout is used as output
```

6 [filters.py](#)

This script creates a configuration file to be sent to the **WebdynEasy** to activate filter upon medium and/or manufacturers.

See script: <https://www.webdyn.com/download/filters.py>

```
usage:
    filters.py "ITR,SAP,ZRI" "03,07,0c" ouputfileName.bson only

medium filter: filters.py "" "03,07,0c" ouputfileName.bson
only manufacturer filter: filters.py "ITR,SAP,ZRI" "" ouputfileName.bson
```

7 [whiteList.py](#)

This script creates, from files containing meter list, a bson Whitelist file to be sent to the WebdynEasy. (see manual §4.2.2.1 page 44)

Naming the output file after this format : “<uid>-wlFilter.bson” will allow you to pass it directly to the /INBOX/uid folder for this whitelist to be taken into account after next connection.

See script: <https://www.webdyn.com/download/whiteList.py>

```
usage:
    whiteList.py ouputfileName.bson file1.wl file2.wl ...

the input files contant:
    first line:      Manufacturer code (3 letters ex: WDN)
    second line:    Generation or versioin code in hexadecimal (1 byte)
    third line:     Medium code in hexadecimal (1 byte)
    forth line:     List of the radio ID of the meters separated by comma
```

Example:

```
python3 whiteList.py out-wlFilter.bson sap.wl itr.wl
```

With sap.wl:

```
SAP
c0
01
20992a41,2099293d,209929c7,20992a3d,209929c1,20992a36
```

and itr.wl:

```
ITR
1e
03
11000310,11000311,11000312,11000313
```

Result: out-wlFilter.bson (converted to json):

```
{
  "wlFilter": [
    {
      "id": {
        "$binary": "MExBKpkgwAE=",
        "$type": "00"
      }
    },
    {
      "id": {
        "$binary": "MEw9KZkgwAE=",
```

```
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "MEzHKZkgwAE=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "MEw9KpkgwAE=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "MEzBKZkgwAE=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "MEw2KpkgwAE=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "kiYQAwARHgM=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "kiYRAwARHgM=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "kiYSAwARHgM=",
    "$type": "00"
  }
},
{
  "id": {
    "$binary": "kiYTAwARHgM=",
    "$type": "00"
  }
}
],
"crc": -1381163023
}
```

8 [csv2whiteList.py](#)

This script creates, from a csv file, a bson Whitelist file to be sent to the WebdynEasy. (see manual §4.2.2.1 page 44)

csv file must have the same format as the one obtained when using data2csv.py (§5), whatever the option used for date format (-d) or manufacturer format (-m).

This allows you to retrieve a first data file from site, convert it thanks to the script data2csv.py (§5) modify the csv file as wanted (remove telegrams from unwanted devices eventually add some) and generate a dedicated WhiteList to the specific site.

Usefull tip: magnet option 1 (§4.2.1.1) will trigger a Diagnostic command when magnet is used to go from storage mode to run mode. This diagnostic command is retrieving radio telegrams before doing the connection to the FTP.

Naming the output file after this format: “<uid>-wFilter.bson” will allow you to pass it directly to the /INBOX/uid folder for this whitelist to be taken into account after next connection.

See script:

<https://www.webdyn.com/download/csv2whiteList.py>

```
usage:
```

```
    csv2whiteList.py inputfile.csv outputfile.bson
```

```
if the first arg is '-' stdin is used as input if  
the second arg is '-' stdout is used as output
```

Example:

```
csv2whiteList.py 456383-1606379932-data.csv 456383-wlFilter.bson
```

with 456383-1606379932-data.csv:

```
date, rssi, length, C field, manufacturer, radio id, generation, medium, CI field, payload
```

```
2000-01-01 01:01:09, -  
70, 138, 44, SON, 20591301, 31, 01, 7a, 2d00a025822adbcf1552ed4711f34792e90
```

```
2000-01-01 01:01:10, -  
71, 138, 44, SON, 20591302, 31, 01, 7a, 2d00a025822adbcf1552ed4711f34792e90
```


result:

```
{
  "wlFilter": [
    {
      "id": {
        "$binary": "7k0BE1kgMQE=",
        "$type": "00"
      }
    },
    {
      "id": {
        "$binary": "7k0CE1kgMQE=",
        "$type": "00"
      }
    }
  ],
  "crc": 1829404939
}
```