VirtualWisdom Import Utilities

Index

Overview

The VirtualWisdom Import Utilies consist of the following three scripts:

1. vw_csv_nicknames_to_json
2. vw_csv_relations_to_json
3. vw_import_entities

vw_csv_nicknames_to_json generates importable JSON from a CSV file containing WWN to nickname (alias) mappings.

Input format

WWN,nickname

Example:

```
248d00059b2b78c0,hba1port1
5001438002b0dbb4,hba1port2
```

vw_csv_relations_to_json generats importable JSON from a CSV file containing entity definitions.

Input Format

Type,Name,Tags,Item1,Item2,...,ItemN

Type is one of: application, hba, host, storagearray, storagecontroller, iomodule.

Tags is a semicolon-separated list of words.

Example:

```
hba,hba1,tag1;tag2,hba1port1,hba1port2
hba,hba2,,hba2port1,hba2port2
host,host1,tag3;tag4,hba1
host,host2,,hba2
application,app1,tag5,host1,host2
```

vw_import_entities imports JSON into VirtualWisdom.

See the Usage section below for information on how to use each script.

Support

These scripts are provided "as is" and are not supported by VirtualInstruments.

Installation

These scripts require Python 3.7+ (64-bit), two Python packages, 'click' and 'requests', and a version of OpenSSL that supports TLSv1.2. The easiest and recommended method for setting up to run the scripts is as follows.

After item 1 below, the remaining command line demonstrations will be done on a Unix-like system, macOS in this case.

1. Create a virtual environment:

```
Unix/Linux/macOS              Windows

$ python3 -m venv venv        c:\<dir> py -3 -m venv venv
$ venv/bin/activate           c:\<dir> venv\Scripts\activate
(venv) $                      (venv) c:\<dir>
```

2. Install the package.

```
(venv) $ pip install https://github.com/Virtual-Instruments/vwimportutils/archive/v1.0.1-1.zip
```

3. Validate that the package installed correctly.

```
(venv) $ vw_import_entities --help
```

vw_csv_nicknames_to_json

```
Usage: vw_csv_nicknames_to_json.py [OPTIONS] CSV_IN JSON_OUT

  This script generates an importable JSON file from a CSV file containing
  WWN to nickname (alias) mappings.

  Input is a CSV file formatted as follows:

  WWN1,nickname1
  WWN2,nickname2

  Output is a JSON file that can be imported into VirtualWisdom, either via
  the UI or via the command line using the vw_import_entities script.

  The --etype (-t) argument must be either hostport or storageport.

  The command is pipeable; simply replace either the input file, output
  file, or both with a dash (-).

  Examples (Linux/macOS/Unix):

  (venv) $ vw_csv_nicknames_to_json -t hostport aliases.csv import.json
  (venv) $ cat aliases.csv | vw_csv_nicknames_to_json -t hostport - - |
  vw_import_entities ...

Options:
  -t, --etype TEXT
  --help            Show this message and exit.
```

vw_csv_relations_to_json

```
Usage: vw_csv_relations_to_json [OPTIONS] CSV_IN JSON_OUT

  This script generates an importable JSON file from a CSV file containing
  entity definitions.

  Input is a CSV file formatted as follows:

  Type,Name,Tags,Item1,Item2,...,ItemN

  Type is one of: application, hba, host, storagearray, storagecontroller,
  iomodule.

  Tags is a semicolon-separated list of words.

  Example

  hba,hba1,tag1;tag2;tag3,hba1port1,hba1port2
  hba,hba2,,hba2port1,hba2port2
  host,host1,tag4;tag5,hba1
  host,host2,,hba2
  application,app1,tag6;tag7,host1,host2

  Output is a JSON file that can be imported into VirtualWisdom, either via
  the UI or via the command line using the vw_import_entities script.
  The command is pipeable; simply replace either the input file, output
  file, or both with a dash (-).

  Examples (Linux/macOS/Unix):

  (venv) $ vw_csv_relations_to_json relations.csv import.json

  (venv) $ cat relations.csv | vw_csv_relations_to_json - - |
  vw_import_entities ... -

Options:
  --help  Show this message and exit.
```

vw_import_entities

```
Usage: vw_import_entities [OPTIONS] JSON_IN

  This script imports entities (or aliases) into VirtualWisdom. It does so
  using VW's Public REST API. As such, it requires two things: (1) a token
  generated from the VW UI (see Ch. 8 in the VW User Guide), and (2) a REST
  API SDK license, properly installed in the target VW Appliance.

  Input is a properly constructed JSON import file (see Ch. 8 in the VW User
  Guide).

  The command is pipeable; simply replace the input file with a dash (-).

  Examples (Linux/macOS/Unix):

  (venv) $ vw_import_entities -h 10.20.30.40 -t <token> entities.json

  (venv) $ cat aliases.csv | vw_csv_nicknames_to_json - - |
  vw_import_entities -h 10.20.30.40 -t <token> -

  If your input file has any errors, they will be listed to the screen.

  You can set two environment variables to simplify the use of this script:

  On Mac/Linux/Unix:

  export VI_IPADDR=10.20.30.40
  export VI_TOKEN=<token>

  On Windows:

  set VI_IPADDR=10.20.30.40
  set VI_TOKEN=<token>

  Doing so eliminates the need to use the -h and -t options.

Options:
  -h, --host TEXT
  -t, --token TEXT
  -F, --force
  --help            Show this message and exit.
```