txt2xls Documentation

Release 0.2.2

Alisue

March 03, 2016

Contents

CHAPTER 1

Synopsis

txt2xls [*options*] -o <*output filename*> [*pathnames ...*]

CHAPTER 2

Description

txt2xls is a tool for automatic convertion for raw text files into a single Microsoft Excel file.

It use maidenhair parser and loader plugins for loading raw text files. It means that **txt2xls** can load any formats of raw text files when there are plugins suite for the format.

CHAPTER 3

Options

-h, --help show this help message and exit

-v, --version show program's version number and exit

--raise-exception If it is specified, raise exceptions.

-o OUTFILE, --outfile OUTFILE An output filename without extensions. The required filename extension will be automatically determined from an output format.

3.1 Reading options

-p PARSER, --parser PARSER A maidenhair parser name which will be used to parse the raw text data.

-l LOADER, --loader LOADER A maidenhair loader name which will be used to load the raw text data.

-u USING, --using USING A colon (:) separated column indexes. It is used for limiting the reading columns.

3.2 Unite options

--unite Join the columns of classified dataset with respecting *–unite-basecolumn*. The dataset is classified with *–unite-function*.

--unite-basecolumn UNITE_BASECOLUMN An index of columns which will be used as a base column for regulating data point region.

--unite-function UNITE_FUNCTION A python script file path or a content of python lambda expression which will be used for classifing dataset. If it is not spcified, a filename character before period (.) will be used to classify. See **Function expression** section below.

3.3 Classify options

--classify Classify dataset with –classify-function. It will influence the results of –relative and –baseline.

--classify-function CLASSIFY_FUNCTION A python script file path or a content of python lambda expression which will be used for classifing dataset. If it is not specified, a filename character before the last underscore (_) will be used to classify. See **Function expression** section below.

3.4 Relative options

--relative If it is True, the raw data will be converted to relative data from the specified origin, based on the specified column. See *–relative-origin* and *–relative-basecolumn* also.

--relative-origin RELATIVE_ORIGIN A dataset number which will be used as an orign of the relative data. It is used with *–relative* option.

--relative-basecolumn RELATIVE_BASECOLUMN A column number which will be used as a base column to make the data relative. It is used with *–relative* option.

3.5 Baseline options

--baseline If it is specified, the specified data file is used as a baseline of the dataset. See *–baseline-basecolumn* and *–baseline-function* also.

--baseline-basecolumn BASELINE_BASECOLUMN A column index which will be proceeded for baseline regulation. It is used with *–baseline* option.

--baseline-function BASELINE_FUNCTION A python script file path or a content of python lambda expression which will be used to determine the baseline value from the data. *columns* and *column* variables are available in the lambda expression. See **Function expression** section below.

3.6 Peakset options

--peakset-method {argmin, argmax} A method to find peak data point.

--peakset-basecolumn PEAKSET_BASECOLUMN A column index which will be used for finding peak data point.

--peakset-where-function PEAKSET_WHERE_FUNCTION A python script file path or a content of python lambda expression which will be used to limit the range of data points for finding. peak data point. *data* is available in the lambda expression. See **Function expression** section below.

CHAPTER 4

Function expression

There are four types of function expressions and there are identified by leading function type indicator; characters before the first colon (:).

4.1 lambda function

A function expression starts from `lambda:` indicate the lambda function expression and the body (string after the `lambda:`) indicate the body of the lambda function. The lambda function will recieve `*args` and `**kwargs` arguments when it is called so you can write a lambda function which return the first argument as:

```
lambda:args[0]
```

The function expression above will be converted to:

```
lambda *args, **kwargs: args[0]
```

4.2 regex function

A function expression starts from `regex:` indicate the regex function expression and the body indicate the regular expression pattern string. The regex function will recieve `data` which first item indicate the filename of the data (a row text filename) and the function check the filename with the specified regular expression pattern. This function is mainly used for classification function such as `--unite-function` or `--classify-function`.

If the regular expression pattern has group patterns, it will return the first group as a classification string. It it does not have group patterns, it will return the entire match string. If nothing can be mathced in the specified filename, entire filename will be returned as a classification string.

4.3 file function

A function expression starts from `file:` indicate the file function expression and the body indicate the path of the python script. The python script will be loaded and it's `__call__(data)` function will be used as a function. It the python script does not have the function, it raise `ImportError`.

4.4 builtin function

A function expression starts from `builtin:` is a shortcut alias of file function which points to builtin python script files. Currently four builtin scripts are available (`baseline_function`, `classify_function`, `unite_function`, and `where_function`).

CHAPTER 5

Preference

You can create configure file as `~/.config/txt2xls/txt2xls.cfg` (Linux), `~/.txt2xls.cfg` (Mac), or `%APPDATA%\txt2xls\txt2xls.cfg` (Windows).

The default preference is equal to the configure file as below:

```
[default]
raise_exception = False

[reader]
parser = 'parsers.PlainParser'
loader = 'loaders.PlainLoader'
using = None

    [[classify]]
    enabled = False
    function = 'builtin:classify_function'

    [[unite]]
    enabled = False
    function = 'builtin:unite_function'
    basecolumn = 0

    [[relative]]
    enabled = False
    origin = 0
    basecolumn = 1

    [[baseline]]
    enabled = False
    function = 'builtin:baseline_function'
    basecolumn = 1

[writer]
default_filename = 'output.xls'

    [[peakset]]
    method = 'argmax'
    basecolumn = -1
    where_function = 'builtin:where_function'
```

CHAPTER 6

API documentation

6.1 txt2xls Package

6.1.1 txt2xls Package

6.1.2 args Module

txt2xls.args.**parse_args**(*args=None*)

txt2xls.args.**parse_using**(*value*)

6.1.3 compat Module

6.1.4 conf Module

txt2xls.conf.**get_user_config_filename**(*appname*)

Get user config filename.

It will return operating system dependent config filename.

Parameters **appname** (*string*) – An application name used for filename

Returns A filename of user configuration.

Return type string

txt2xls.conf.**overwrite_conf_with_args**(*conf*, *args*)

txt2xls.conf.**parse_conf**(*appname*, *args=None*)

6.1.5 console Module

txt2xls.console.**txt2xls**(*args=None*)

6.1.6 `utils` Module

6.1.7 Subpackages

function Package

`function` Package

txt2xls.function.**create_file_function**(*body*)
Create function from specified python script file.

The python script file has to have `__call__(data)__` function.

Parameters **body** (*string*) – A filename of the python script

Returns A function

Return type *function*

txt2xls.function.**create_lambda_function**(*body*)
Create lambda function with partial code.

The partial code (`body`) will be put inside of lambda expression as:

```
lambda *args, **kwargs: <BODY>
```

Parameters **body** (*string*) – A body part of lambda expression

Returns A lambda function

Return type *function*

txt2xls.function.**create_regex_function**(*body*)
Create function from regex pattern.

It is used for creating a function for classification.

Parameters **body** (*string*) – A regex pattern

Returns A function which will return string for classification

Return type *function*

txt2xls.function.**parse_function**(*function*, *default_type='lambda'*)
Parse function expression which start from function type (e.g. 'lambda:')

The function expression has to be start from 'lambda:', 'file:', 'regex:', or 'builtin:' (shortcut alias of 'file:' type for builtin functions). If no function type is specified, the function will be treated as `default_type`.

Parameters

- **function** (*string*) – A function expression
- **default_type** (*string*) – A default function expression type

Returns A function

Return type *function*

Raises `AttributeError` – Raise when unknown function type has been specified.

Subpackages

builtin Package

baseline_function Module

txt2xls.function.builtin.baseline_function.**default_baseline_function**(*data*, *column*)

classify_function Module

txt2xls.function.builtin.classify_function.**default_classify_function**(*data*)

A default classify_function which recieve *data* and return filename without characters just after the last underscore

```
>>> # [<filename>] is mimicking `data`
>>> default_classify_function(['./foo/foo_bar_hoge.piyo'])
'./foo/foo_bar.piyo'
>>> default_classify_function(['./foo/foo_bar.piyo'])
'./foo/foo.piyo'
>>> default_classify_function(['./foo/foo.piyo'])
'./foo/foo.piyo'
>>> default_classify_function(['./foo/foo'])
'./foo/foo'
```

unite_function Module

txt2xls.function.builtin.unite_function.**default_unite_function**(*data*)

A default unite_function which recieve *data* and return filename without middle extensions

```
>>> # [<filename>] is mimicking `data`
>>> default_unite_function(['./foo/foo.bar.hoge.piyo'])
'./foo/foo.piyo'
>>> default_unite_function(['./foo/foo.piyo'])
'./foo/foo.piyo'
>>> default_unite_function(['./foo/foo'])
'./foo/foo'
```

where_function Module

reader Package

reader Package

class txt2xls.reader.**Reader**(*conf*)

Bases: object

read(*pathname*, *fail_silently*)

utils Module

writer Package

writer Package

class txt2xls.writer.**Writer**(*conf*)

Bases: object

write(*collection*, *filename=None*, *fail_silently=False*)

utils Module

txt2xls.writer.utils.**ensure_iterable**(*axis*)

Ensure the axis is iterable (XY array)

```
>>> axis1 = [
...     [0, 1, 2],
...     [3, 4, 5],
...     [6, 7, 8],
... ]
>>> assert axis1 == ensure_iterable(axis1)
>>> axis21 = [0, 1, 2]
>>> axis22 = [[0], [1], [2]]
>>> assert axis22 == ensure_iterable(axis21)
```

txt2xls.writer.utils.**get_sheet_name**(*filename*)

Return extension and parent directory stripped filename which is used as a sheet name

Parameters **filename** (*string*) – A file path

Returns A extension and parent directory stripped filename

Return type filename

Examples

```
>>> get_sheet_name('./foo/bar/hogehoge.piyo')
'hogehoge'
>>> len(get_sheet_name("*"*100))
31
```

txt2xls.writer.utils.**prefer_alphabet**(*i*)

Convert an integer to an alphabet if it is within 0 to 51.

```
>>> prefer_alphabet(0)
'A'
>>> prefer_alphabet(25)
'Z'
>>> prefer_alphabet(26)
'a'
>>> prefer_alphabet(51)
'z'
>>> prefer_alphabet(100)
'100'
```

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

t

C

D

E

G

O

P

R

T

W