
Verace Documentation

Release 0.4.1

Jeff Rimko

Aug 23, 2019

Contents

1	VerChecker	3
2	Custom Check Functions	5
2.1	File Functions	5
2.2	Line Functions	5
	Index	7

This is the main documentation for Verace ([GitHub](#), [PyPI](#)), a Python library to aid checking strings in files.

CHAPTER 1

VerChecker

This class is used to create objects that will check/update strings in files:

class verace.**VerChecker** (*name, root*)

include (*path, func=None, opts=None, updatable=True, **kwargs*)

Includes a file to check.

Params:

- *path* (str) - Path to a file to check.
- *func* (function/(tuple,str)) - Either a (func,type) tuple or a function (type defaults to 'file'). Valid type values are 'fileline'. The type controls whether the file path or individual lines will be passed to the function. A file function may return either single VerInfo or a list of VerInfos. A line function may return a string.
- *opts* (dict) - Options to pass to the check function. Any additional keyword args will be included.
- *updatable* (bool) - If true, string can be updated using *update()*.

prompt ()

Shows the standard prompt for handling checked strings.

run (*verbose=True*)

Runs checks on all included items, reports any inconsistencies. Returns string if consistent else None.

string ()

Returns the string if *run()* found no inconsistencies, otherwise None is returned. Always calls *run()*.

update (*newstr*)

Updates all associated strings to the given new string. Use caution as this will modify file content! Returns number of strings updated.

Custom Check Functions

A function can be provided to `VerChecker.include()`. The function will be used to parse the file for the target string.

If a tuple/list is provided as the `func` argument, the first value must be a function and the second value must be the string `'file'` (default) or `'line'`.

If no explicit function is provide, the following file function will be used:

`verace.check_basic(path, match='version', splits=None, single=True)`

Basic check function. Iterates through files lines until the *match* string is found. The matching line will be split using the list of (characters,index) tuples/lists from *splits*. Can either return a single (first) result or all results in the file. **NOTE:** *splits* must be a list of tuples/lists!

2.1 File Functions

File functions must handle the string search through the entire target file. The first argument must be the file path and the function must return either a single `VerInfo` object or a list of `VerInfo` objects:

`verace.VerInfo = <class 'verace.VerInfo'>`

Contains information for a single checked item.

Use this convenience function to iterate through the lines in a given file:

`verace.readlines(fpath)`

Generator that reads the file at the given path line by line yielding (number,text) for each.

2.2 Line Functions

Line functions will be provided each line from the file one at a time. If the target string is found, simply return it. Otherwise return `None` (default return).

C

`check_basic()` (*in module verace*), 5

I

`include()` (*verace.VerChecker method*), 3

P

`prompt()` (*verace.VerChecker method*), 3

R

`readlines()` (*in module verace*), 5

`run()` (*verace.VerChecker method*), 3

S

`string()` (*verace.VerChecker method*), 3

U

`update()` (*verace.VerChecker method*), 3

V

`VerChecker` (*class in verace*), 3

`VerInfo` (*in module verace*), 5