
smartfilesorter Documentation

Release 0.2.0

Jason Short

September 14, 2014

1	Smart File Sorter	3
1.1	Features	3
2	Installation	5
3	Usage Example	7
3.1	Basic Example	7
3.2	Advanced Example	7
4	Match Plugins	11
4.1	file-extension-is	11
4.2	filename-contains	11
4.3	filename-ends-with	12
4.4	filename-matches	12
4.5	filename-starts-with	12
5	Action Plugins	15
5.1	move-to	15
5.2	print-file-info	15
5.3	rename-to	15
5.4	stop-processing	16
6	Contributing	17
6.1	Types of Contributions	17
6.2	Get Started!	18
6.3	Pull Request Guidelines	18
6.4	Tips	19
7	Credits	21
7.1	Development Lead	21
7.2	Contributors	21
8	History	23
9	0.2.0 (2014-09-15)	25
10	0.1.0 (2014-09-14)	27
11	Indices and tables	29

Contents:

Smart File Sorter

Rule based file moving and renaming tool

- Free software: BSD license
- Documentation: <https://smartfilesorter.readthedocs.org>.

1.1 Features

- Moves/renames files based on rules defined in a YAML configuration file.
- Automatically renames a file if it already exists in the destination directory by appending a sequence number to the filename. (file.txt, file_001.txt, file_002.txt, etc)
- Easy to extend with new match or action rules

Installation

At the command line:

```
$ easy_install SmartFileSorter
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv SmartFileSorter  
$ pip install SmartFileSorter
```

Usage Example

- Basic Example
- Advanced Example

3.1 Basic Example

To use SmartFileSorter, create a rule file. For example:

```
# test.yml
- name: Move Logs
  match:
    - file-extension-is: .log
  action:
    - move-to: /archive
```

Then run the `sfp` command. In this case, it will process the rules in the `test.yml` file against every file in the `/tmp` directory, without actually performing any actions. Assuming there are two files with the `.log` extension in `/tmp` (`test1.log` and `test2.log`), the output would look like this:

```
$ sfs test.yml /tmp --dry-run
Running with --dry-run parameter. Actions will not be performed.
Move Logs: test1.log - Match
Move Logs: test2.log - Match
Files matched: 2/10
```

And to actually move the files, run without the `--dry-run` parameter:

```
$ sfs test.yml /tmp
Move Logs: test1.log - Match
Move Logs: test2.log - Match
Files matched: 2/10
```

The two files would be moved to the directory `/archive`.

3.2 Advanced Example

In this example, we'll move and rename log files in the `/tmp/` directory in to the `/archive/` directory by year. Assume that the log files are named `test-YY-MM-DD.log` and we want them to be named `YYYY-MM-DD.log`

```
# test2.yml
- name: Move and Rename 2014 Logs
  match:
    - filename-starts-with: test-14
    - file-extension-is: .log
  action:
    - rename-to:
      match: ^test-14
      replace-with: 2014
    - move-to: /archive/2014/

- name: Move and Rename 2013 Logs
  match:
    - filename-starts-with: test-13
    - file-extension-is: .log
  action:
    - rename-to:
      match: ^test-13
      replace-with: 2013
    - move-to: /archive/2013/

- name: Move and Rename 2012 data
  match:
    - filename-starts-with: test-12
    - file-extension-is: .log
  action:
    - rename-to:
      match: ^test-12
      replace-with: 2012
    - move-to: /archive/2012/
```

Use `--dry-run` to see what files would be affected:

```
$ ls /tmp
test-13-05-01.log
test-14-01-01.log
test-14-01-02.log

$ sfs test2.yml /tmp --dry-run
Running with --dry-run parameter. Actions will not be performed.
Move Logs: test-13-05-01.log - Match
Move Logs: test-14-01-01.log - Match
Move Logs: test-14-01-02.log - Match
Files matched: 3/3
```

And to actually move the files, run without the `--dry-run` parameter:

```
$ sfs test2.yml /tmp
Move Logs: test-13-05-01.log - Match
Move Logs: test-14-01-01.log - Match
Move Logs: test-14-01-02.log - Match
Files matched: 3/3
```

Here are the results:

```
$ ls -R /archive
/archive/2012:

/archive/2013:
```

```
test-2013-05-01.log
```

```
/archive/2014:
```

```
test-2014-01-01.log
```

```
test-2014-01-02.log
```

Match Plugins

Match Plugins indicate if a given file matches the rule it defines. For example, if the file has a certain extension, or if the file name starts with a certain character. The available match plugins are:

- `file-extension-is`
- `filename-contains`
- `filename-ends-with`
- `filename-matches`
- `filename-starts-with`

4.1 file-extension-is

Matches if the file's extension is equal to the given extension. Not case sensitive. More than one extension may be given, space separated.

```
match:  
- file-extension-is: .log .txt
```

Would match:

- `test.log`
- `test.txt`
- `a.file.with.multiple.periods.txt`

But not:

- `test.jpg`

4.2 filename-contains

Matches if the filename (without extension) contains the given string. Not case sensitive.

```
match:  
- filename-contains: foo
```

Would match:

- `this_file_has_foo.log`

- foo_and_more_foo.log

But not:

- test.jpg
- test.foo

4.3 filename-ends-with

Matches if the filename (without extension) ends with the given string. Not case sensitive.

```
match:
- filename-ends-with: bar
```

Would match:

- foo_and_bar.log
- bar.log

But not:

- bar_test.jpg
- test.bar

4.4 filename-matches

Matches if the filename (without extension) matches the given regular expression. Is case sensitive unless the regex says otherwise. See <https://docs.python.org/3/howto/regex.html> for more information on regular expressions.

```
match:
- filename-matches: ^\d\d\d\d.
```

Would match:

- 1234.log
- 7890_and_other_things.log

But not:

- abc1234.jpg
- 12.log

4.5 filename-starts-with

Matches if the filename (without extension) starts with the given string. Not case sensitive.

```
match:
- filename-starts-with: abc
```

Would match:

- abcdefg.log
- abc.log

- AbCdEf.txt

But not:

- bcdEf.jpg

Action Plugins

Action Plugins tell Smart File Sorter what to do with a file if it matches all the rules in that ruleset. For example, rename or move the file. The available action plugins are:

- `move-to`
- `print-file-info`
- `rename-to`
- `stop-processing`

5.1 move-to

Moves the file to the given directory. If a file with the same name already exists, the current file will have a numbered suffix appended. For example, `abc.log` would be renamed to `abc_001.log`.

```
action:
- move-to: /archive/
```

Would move the file to the `/archive/` directory.

5.2 print-file-info

Prints information on the file. Mostly used for testing.

```
action:
- print-file-info
```

5.3 rename-to

In the filename, replaces the match value (a regular expression) with the replace-with value. Is case sensitive

```
action:
- rename-to:
  match: ^\d\d\d\d
  replace-with: abcd
```

Would change:

- 1234_and_things.txt to abcd_and_things
- 7980.txt to abcd.txt

action:

```
- rename-to:  
  match: XYZ  
  replace-with: 123
```

Would change:

- XYZ.txt to 123.txt
- xyz_and_XYZ.txt to xyz_and_XYZ.txt

5.4 stop-processing

This rule stops all further processing on the current file.

action:

```
- stop-processing
```

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at <https://github.com/jashort/SmartFileSorter/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

6.1.4 Write Documentation

SmartFileSorter could always use more documentation, whether as part of the official SmartFileSorter docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/jashort/SmartFileSorter/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.2 Get Started!

Ready to contribute? Here's how to set up *SmartFileSorter* for local development.

1. Fork the *SmartFileSorter* repo on GitHub.

2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/SmartFileSorter.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv smartfilesorter
$ cd smartfilesorter/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 smartfilesorter tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

6.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/jashort/smartfilesorter/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_smartfilesorter
```

Credits

7.1 Development Lead

- Jason Short <jason@sheersky.com>

7.2 Contributors

None yet. Why not be the first?

History

0.2.0 (2014-09-15)

- Fixed badge link in README.rst
- Updated documentation with examples and added plugin description and usage examples

0.1.0 (2014-09-14)

- First release on PyPI.

Indices and tables

- *genindex*
- *modindex*
- *search*