
file*readbackwardsDocumentation*

Release 3.0.0

Robin Robin

Mar 29, 2023

Contents

1	file_read_backwards	3
1.1	Features	3
1.2	Usage Examples	3
1.3	Credits	4
2	Installation	5
2.1	Stable release	5
2.2	From sources	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	11
4.4	Tips	11
5	Indices and tables	13

Contents:

Memory efficient way of reading files line-by-line from the end of file

- Free software: MIT license
- Documentation: <https://file-read-backwards.readthedocs.io>.

1.1 Features

This package is for reading file backward line by line as unicode in a memory efficient manner for both Python 2.7 and Python 3.

It currently supports ascii, latin-1, and utf-8 encodings.

It supports “\r”, “\r\n”, and “\n” as new lines.

1.2 Usage Examples

Another example using *python3.11*:

```
from file_read_backwards import FileReadBackwards

with FileReadBackwards("/tmp/file", encoding="utf-8") as frb:

    # getting lines by lines starting from the last line up
    for l in frb:
        print(l)
```

Another way to consume the file is via *readline()*, in *python3.11*:

```
from file_read_backwards import FileReadBackwards

with FileReadBackwards("/tmp/file", encoding="utf-8") as frb:

    while True:
        l = frb.readline()
        if not l:
            break
        print(l, end="")
```

1.3 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.

2.1 Stable release

To install `file_read_backwards`, run this command in your terminal:

```
$ pip install file_read_backwards
```

This is the preferred method to install `file_read_backwards`, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for `file_read_backwards` can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/RobinNil/file_read_backwards
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/RobinNil/file_read_backwards/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

Please see *file_read_backwards*.

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

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/RobinNil/file_read_backwards/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.

4.1.2 Fix Bugs

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

4.1.3 Implement Features

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

4.1.4 Write Documentation

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

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/robin81/file_read_backwards/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 :)

4.2 Get Started!

Ready to contribute? Here's how to set up `file_read_backwards` for local development.

1. Fork the `file_read_backwards` repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/file_read_backwards.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 file_read_backwards
$ cd file_read_backwards/
$ 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 file_read_backwards tests
$ python setup.py test or 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.

4.3 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.

4.4 Tips

To run a subset of tests:

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


CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`