xreshaper Documentation

Release 0.1.0+11.g43512c2

Experimental Development Team

Feb 13, 2019

Contents:

PyReshaper-like operation with Xarray. See documentation for more information.

CHAPTER 1

Installation

1.1 Stable release (COMING SOON!)

To install xreshaper, run this command in your terminal:

```
$ pip install xreshaper
```

This is the preferred method to install xreshaper, 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.

1.2 From sources

You can install specific git commit/tag with pip by running:

```
$ pip install git+git://github.com/NCAR/xarray-pyreshaper.git@0.1.0
```

The sources for xreshaper can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/NCAR/xarray-pyreshaper
```

Or download the tarball:

```
$ curl  -OL https://github.com/NCAR/xarray-pyreshaper/tarball/master
```

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

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

CHAPTER 2

Usage

2.1 Command Line Tool

Xreshaper provides a convenient command line tool via xreshaper-run:

```
$ xreshaper-run --help
Usage: xreshaper-run [OPTIONS]

Options:
--version                 Show the version and exit.
--engine TEXT             Engine to use when reading/writing files.
                         [default: netcdf4]
--input-directory TEXT    Directory in which time-slices files are located
                         [default: ]
--output-directory TEXT   Directory in which time-series files will be saved
                         [default: ]
--output-prefix TEXT      String prefix for all output files.  The output
                         file will be named according to the rule:
                         output_prefix + variable_name + output_suffix
                         [default: tseries.]
--output-suffix TEXT      String suffix for all output files.  The output
                         file will be named according to the rule:
                         output_prefix + variable_name + output_suffix
                         [default: .nc]
--help                    Show this message and exit.
```

CHAPTER 3

Contributing to xreshaper

Contributions are highly welcomed and appreciated. Every little help counts, so do not hesitate!

The following sections cover some general guidelines regarding development in xreshaper for maintainers and contributors. Nothing here is set in stone and can't be changed. Feel free to suggest improvements or changes in the workflow.

Contribution links

3.1 Feature requests and feedback

Do you like xreshaper? Share some love on Twitter or in your blog posts!

We'd also like to hear about your propositions and suggestions. Feel free to submit them as issues and:

- Explain in detail how they should work.
- Keep the scope as narrow as possible. This will make it easier to implement.

3.2 Report bugs

Report bugs for xreshaper in the issue tracker.

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, specifically the Python interpreter version, installed libraries, and xreshaper version.
- Detailed steps to reproduce the bug.

If you can write a demonstration test that currently fails but should pass (xfail), that is a very useful commit to make as well, even if you cannot fix the bug itself.

3.3 Fix bugs

Look through the GitHub issues for bugs.

Talk to developers to find out how you can fix specific bugs.

3.4 Write documentation

xreshaper could always use more documentation. What exactly is needed?

- More complementary documentation. Have you perhaps found something unclear?
- Docstrings. There can never be too many of them.
- Blog posts, articles and such – they're all very appreciated.

You can also edit documentation files directly in the GitHub web interface, without using a local copy. This can be convenient for small fixes.

Note:

Build the documentation locally with the following command:

```
$ conda env update -f ci/environment-dev-3.7.yml
$ cd docs
$ make html
```

The built documentation should be available in the `docs/_build/`.

3.5 Preparing Pull Requests

1. Fork the xreshaper GitHub repository. It's fine to use `xreshaper` as your fork repository name because it will live under your user.
2. Clone your fork locally using git and create a branch:

```
$ git clone git@github.com:YOUR_GITHUB_USERNAME/xreshaper.git
$ cd xreshaper
# now, to fix a bug or add feature create your own branch off "master":

$ git checkout -b your-bugfix-feature-branch-name master
```

If you need some help with Git, follow this quick start guide: https://git.wiki.kernel.org/index.php/QuickStart

3. Install pre-commit and its hook on the xreshaper repo:

```
$ pip install --user pre-commit
$ pre-commit install
```

Afterwards `pre-commit` will run whenever you commit.

https://pre-commit.com/ is a framework for managing and maintaining multi-language pre-commit hooks to ensure code-style and code formatting is consistent.

4. Install dependencies into a new conda environment:

```
$ conda env update -f ci/environment-dev-3.7.yml
```

5. Run all the tests

Now running tests is as simple as issuing this command:

```
$ conda activate xreshaper-dev
$ pytest --junitxml=test-reports/junit.xml --cov=./
```

This command will run tests via the "pytest" tool against Python 3.7.

6. You can now edit your local working copy and run the tests again as necessary. Please follow PEP-8 for naming.

When committing, `pre-commit` will re-format the files if necessary.

7. Commit and push once your tests pass and you are happy with your change(s):

```
$ git commit -a -m "<commit message>"
$ git push -u
```

8. Create a new changelog entry in `changelog`. The file should be named `<issueid>.<type>`, where *issueid* is the number of the issue related to the change and *type* is one of `bugfix`, `removal`, `feature`, `doc` or `trivial`.

9. Add yourself to `AUTHORS` file if not there yet, in alphabetical order.

10. Finally, submit a pull request through the GitHub website using this data:

```
head-fork: YOUR_GITHUB_USERNAME/xreshaper
compare: your-branch-name

base-fork: NCAR/xreshaper
base: master          # if it's a bugfix or feature
```

CHAPTER 4

Indices and tables

- genindex
- modindex
- search