
PyEnvDiff Documentation

Release 0.3.0

Jeffrey McLarty

Nov 19, 2018

Contents

1	PyEnvDiff	3
1.1	via Command-Line	3
1.2	via Hub-Based Compare	3
1.3	Programmatic Usage	4
1.4	Sooo much room for activities!	4
1.5	Installation	4
1.6	Under the hood	5
1.7	Credits	5
2	Installation	7
2.1	Stable release	7
2.2	From sources	7
3	Usage	9
4	Contributing	11
4.1	Types of Contributions	11
4.2	Get Started!	12
4.3	Pull Request Guidelines	13
4.4	Tips	13
5	Indices and tables	15

Contents:

Python environment comparison tool. Maximized for compatibility between versions 2.6 to 3.8, pypy, operating systems, distributions, and forks! Virtualenv, pyenv, pyenv, conda and system!

1.1 via Command-Line

From the command line, to get information on the current environment:

```
python -m pyenvdiff.info
```

Serialize the information to a file...

```
python -m pyenvdiff.info my_environment.json
```

Switch to another environment (you'll need pyenvdiff installed in both)

```
python -m pyenvdiff.compare my_environment.json
```

Or compare two from any environment

```
python -m pyenvdiff.compare my_environment.json my_other_environment.json
```

1.2 via Hub-Based Compare

An HTTP-based service comes with pyenvdiff, all without dependencies (thanks to a copy of bottle). It stores environment information in RAM, for as long as it runs. This service should be considered alpha-stage.

To launch the built-in hub (server):

```
python -m pyenvdiff.hub
```

Then navigate in your browser to the URL it gives you to see the menu of available options. Before you send information about other environments on the same machine, your options will be limited to just viewing the server's environment.

Samples of the [Hub Landing Page](#) and the [Environment Information Page](#) illustrate the features.

From one or more other environments run:

```
python -m pyenvdiff.post_to_hub
```

A URL will be displayed to view environment information from any machine on the same network.

Navigate back to the base URL, you'll see more options to compare the two environments.

A Sample of the [Environment Diff Page](#) illustrate what the diff can do (if `ghdiff` is installed for the hub).`

1.3 Programmatic Usage

```
>>> from pyenvdiff import Environment
>>> e = Environment()
>>> e.to_file('my_env.json')
>>> o = Environment.from_file('other_env.json')
>>> e == o
True
>>> print(e)
... # prints a dump of the environment details
>>> from pyenvdiff import EnvironmentDiff
>>> ed = EnvironmentDiff(e, o)
>>> print(ed)
... # prints a diff of the two environments
```

1.4 Sooo much room for activities!

- Compare dev, test & prod!
- Works on my machine, strange it doesn't work on yours
- Confirming deployments
- Auditing user desktops, servers, research environments & ecosystems
- Filing (or requesting) bug reports

1.5 Installation

There are no mandatory, nor automatically installing, dependencies. There are optional dependencies which increase.

```
pip install pyenvdiff
```

OR just copy & paste pyenvdiff anywhere on PYTHONPATH

There is one optional dependency. The core functionality doesn't use it. It's only needed for pretty HTML-based comparisons via the web.

```
pip install ghdiff
```

1.6 Under the hood

- Robust and organized object model collects and serializes environment information.
- Zero dependency, pure-python, harmless install! Simply *pip install pyenvdiff* or copy anywhere on PYTHONPATH.
- As-needed import statements, to maximize compatibility across python flavours.
- Favours compatible python-code over succinct or newer-style python-code
- Free software: BSD license
- Documentation: <https://pyenvdiff.readthedocs.io>.

1.7 Credits

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

2.1 Stable release

To install PyEnvDiff, run this command in your terminal:

```
$ pip install pyenvdiff
```

This is the preferred method to install PyEnvDiff, 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 PyEnvDiff can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/jnmclarty/pyenvdiff
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/jnmclarty/pyenvdiff/tarball/master
```

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

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


CHAPTER 3

Usage

To use PyEnvDiff in a project:

```
import pyenvdiff
```


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/jnmclarty/pyenvdiff/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

PyEnvDiff could always use more documentation, whether as part of the official PyEnvDiff 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/jnmclarty/pyenvdiff/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 *pyenvdiff* for local development.

1. Fork the *pyenvdiff* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pyenvdiff.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 pyenvdiff
$ cd pyenvdiff/
$ pip install -r requirements_dev.txt
$ 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 pyenvdiff 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.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/jnmclarty/pyenvdiff/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_pyenvdiff
```


CHAPTER 5

Indices and tables

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