
Swytcher Documentation

Release 0.3.2

Eddie Dunn

Dec 17, 2017

Contents

1	Swytcher	3
1.1	Features	3
1.2	Installation	3
1.3	Usage	4
1.4	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	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
6.1	0.3.0 (2017-09-02)	15
6.2	0.1.0 (2017-03-23)	15
7	Indices and tables	17

Contents:

CHAPTER 1

Swytcher

Automatically switches your keyboard layout based on your active window.

Are you using multiple keyboard layouts? Have you discovered that the US keyboard layout is a lot more suitable for programming than the layout you have to use when writing email or otherwise communicating in your native language?

Does it annoy you that you need to remember to manually switch between layouts when going from VIM to your chat/email/whatever app?

If so, Swytcher will solve your problems.

- Free software: MIT license
- Documentation: <https://swytcher.readthedocs.io>.

1.1 Features

- Switch keyboard layout based on X window class
- Switch keyboard layout based on X window name

Planned features

- Switch layout based on combination of window class and name
- Remember last keyboard layout for each window

1.2 Installation

```
$ pip install --user swytcher
```

1.3 Usage

First, copy the sample configuration:

```
$ swytcher --cpcfg
```

Now you can edit `~/.config/swytcher/config.ini` and add the window classes/names to the corresponding layout section.

layout_primary maps to your first keyboard layout, and *layout_secondary* will map to your second. If you have more than two layouts you will have to add more sections to the config file. Each additional layout section must start with *layout*, and be uniquely named. The order that they have in the config file should map to the order that you have defined your different layouts in.

Once you are done editing your config file, you can start Swytcher, and your keyboard layout will be switched whenever you change focus to a window that corresponds to a window class and/or name in your config file.

Swytcher has been tested with the Gnome Desktop Environment and the i3 window manager. Since Swytcher looks at your X keyboard config it should work with most DEs and WMs.

1.4 Credits

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

2.1 Stable release

To install Swytcher, run this command in your terminal:

```
$ pip install swytcher
```

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

You can either clone the public repository:

```
$ git clone git://github.com/eddie-dunn/swytcher
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/eddie-dunn/swytcher/tarball/master
```

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

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


CHAPTER 3

Usage

To use Swytcher in a project:

```
import swytcher
```


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/eddie-dunn/swytcher/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

Swytcher could always use more documentation, whether as part of the official Swytcher 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/eddie-dunn/swytcher/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 *swytcher* for local development.

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

```
$ git clone git@github.com:your_name_here/swytcher.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 swytcher
$ cd swytcher/
$ 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 swytcher 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 3.5, 3.6, and for PyPy. Check https://travis-ci.org/eddie-dunn/swytcher/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_swytcher
```


5.1 Development Lead

- Eddie Dunn <eddie.dunn@gmail.com>

5.2 Contributors

None yet. Why not be the first?

6.1 0.3.0 (2017-09-02)

Added

- `-version` to CLI interface.
- command for copying sample config to user's confdir.

6.2 0.1.0 (2017-03-23)

- First release on PyPI.

CHAPTER 7

Indices and tables

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