# pycard

*Release 0.0.1*

October 26, 2016

Contents

Contents:

CHAPTER 1

# Overview

## 1.1 pycard

| | |
|---|---|
| docs | |
| tests | |

This script facilitates calling */path/to/picard.jar*. Its purpose is to allow you to call picard without providing or even knowing the full path to the executable far file.

- Free software: MIT license

## 1.2 Usage

To use pycard in a from the command line:

```
$ pycard --help
Usage: pycard [OPTIONS] COMMAND [ARGS]...

  This script facilitates calling `/path/to/picard.jar`.

  Its purpose is to allow you to call picard without providing or even
  knowing the full path to the executable far file.

Options:
  --use-path PATH  Provide a path to override the default picard.jar location.
  --version        Show the version and exit.
  --help           Show this message and exit.

Commands:
  do    runs picard
  help  shows the picard.jar help text
```

```
$ pycard do --help
Usage: pycard do [OPTIONS] [PICARD_ARG]...

  This command actually calls picard.
```

```
  PICARD_ARG          These will be passed directly to picard.

  The help text for picard can be accessed by providing zero PICARD_ARGs.


  Example usages:

      pycard do PicardCommandName OPTION1=value1 OPTION2=value2...
      pycard do --jvm-args '-Xmx6g' PicardCommandName OPTION1=value1 OPTION2=value2...
      pycard do PicardCommandName OPTION1=value1 OPTION2=value2... --jvm-args '-Xmx6g'

Options:
  --jvm-args TEXT  a quoted string that contains the args you wish to pass to
                   the java virtual machine.  [default: '-Xmx2g']
  --help           Show this message and exit.
```

### 1.2.1 Installation

**Danger:** There is a `PyCard` on PyPI, but that is **NOT** this package!
Do **NOT** simply `pip install pycard` or you will end up with that address card program!

```
git clone https://github.com/xguse/pycard.git
cd pycard
pip install .
```

Or via conda:

```
conda install pycard -c http://xguse.github.io/conda-package-repo/pkgs/channel/
```

### 1.2.2 Documentation

https://pycard.readthedocs.org/

### 1.2.3 Development

To run the all tests run:

```
tox
```

CHAPTER 2

# Installation

## 2.1 Via pip

**Danger:** There is a `PyCard` on PyPI but that is **NOT** this package!
Do **NOT** simply `pip install pycard` or you will end up with that address card program!

From github:

```
pip install -e git://github.com/xguse/pycard.git@v0.0.1#egg=pycard
```

or for the very latest:

```
pip install -e git://github.com/xguse/pycard.git@master#egg=pycard
```

or for the very VERY latest:

```
pip install -e git://github.com/xguse/pycard.git@develop#egg=pycard
```

## 2.2 Via conda

Via my personal repo manually:

```
conda install pycard -c http://xguse.github.io/conda-package-repo/pkgs/channel/
```

Or add the repo to your *.condarc* to search the repo whenever you do `conda install`:

```
conda config --add channels http://xguse.github.io/conda-package-repo/pkgs/channel/
conda install pycard
```

CHAPTER 3

# Usage

To use pycard in a from the command line:

```
$ pycard --help
Usage: pycard [OPTIONS] COMMAND [ARGS]...

  This script facilitates calling `/path/to/picard.jar`.

  Its purpose is to allow you to call picard without providing or even
  knowing the full path to the executable far file.

Options:
  --use-path PATH  Provide a path to override the default picard.jar location.
  --version        Show the version and exit.
  --help           Show this message and exit.

Commands:
  do    runs picard
  help  shows the picard.jar help text
```

```
$ pycard do --help
Usage: pycard do [OPTIONS] [PICARD_ARG]...

  This command actually calls picard.

  PICARD_ARG          These will be passed directly to picard.

  The help text for picard can be accessed by providing zero PICARD_ARGs.


  Example usages:

      pycard do PicardCommandName OPTION1=value1 OPTION2=value2...
      pycard do --jvm-args '-Xmx6g' PicardCommandName OPTION1=value1 OPTION2=value2...
      pycard do PicardCommandName OPTION1=value1 OPTION2=value2... --jvm-args '-Xmx6g'

Options:
  --jvm-args TEXT  a quoted string that contains the args you wish to pass to
                   the java virtual machine.  [default: '-Xmx2g']
  --help           Show this message and exit.
```

CHAPTER 4

# Reference

## 4.1 pycard

CHAPTER 5

# Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

## 5.1 Bug reports

When 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.

## 5.2 Documentation improvements

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

## 5.3 Feature requests and feedback

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

## 5.4 Development

To set up *pycard* for local development:

1. Fork pycard on GitHub.
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/pycard.git
```

3. Create a branch for local development:

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

   Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with tox one command:

```
tox
```

5. 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
```

6. Submit a pull request through the GitHub website.

### 5.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`) [1].
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

### 5.4.2 Tips

To run a subset of tests:

```
tox -e envname -- py.test -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

[1] If you don't have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though ...

CHAPTER 6

# Authors

- Gus Dunn - https://github.com/xguse/pycard

CHAPTER 7

# Changelog

## 7.1 0.0.1 (2015-10-13)

- First release on PyPI.

CHAPTER 8

# Indices and tables

- genindex
- modindex
- search