scikit-build Documentation

Release 0.3.0

scikit-build team

September 18, 2016

User guide

scikit-build is an improved build system generator for CPython C extensions. It provides better support for additional compilers, build systems, cross compilation, and locating dependencies and their associated build requirements.

The **scikit-build** package is fundamentally just glue between the *setuptools* Python module and CMake. Currently, the package is available to perform builds in a *setup.py* file. In the future, the project aims to be a build tool option in the currently developing pyproject.toml build system specification.

CHAPTER 1

Installation

1.1 Install package with pip

To install with pip:

```
$ pip install scikit-build
```

1.2 Install from source

To install scikit-build from the latest source, first obtain the source code:

```
$ git clone https://github.com/scikit-build/scikit-build
$ cd scikit-build
```

then install with:

```
$ pip install .
```

or:

```
$ pip install -e .
```

for development.

1.3 Dependencies

1.3.1 Python Packages

The project has a few common Python package dependencies. The runtime dependencies are:

```
wheel==0.29.0
setuptools==26.1.1
```

The build time dependencies (also required for development) are:

```
codecov==2.0.5
coverage==4.2
cython==0.24.1
flake8==3.0.4
```

```
pytest==3.0.2
pytest-cov==2.3.1
pytest-mock==1.2
pytest-runner==2.9
six==1.10.0
```

1.3.2 Compiler Toolchain

The same compiler toolchain used to build the CPython interpreter should also be available. Refer to the CPython Developer's Guide for details about the compiler toolchain for your operating system.

For example, on *Ubuntu Linux*, install with:

```
$ sudo apt-get install build-essential
```

On *Mac OSX*, install XCode to build packages for the system Python.

On Windows, install the version of Visual Studio used to create the target version of CPython

1.3.3 CMake

Download standard CMake binaries for your platform. Alternatively, build CMake from source with a C++ compiler if binaries are not available for your operating system.

CHAPTER 2

Usage

2.1 Basic usage

To use scikit-build in a project, place the following in your project's *setup.py* file:

```
# This line replaces 'from setuptools import setup'
from skbuild import setup
```

Now, your project will use scikit-build instead of setuptools.

2.2 Controlling CMake using scikit-build

Alternatively, you can drive CMake more directly yourself using scikit-build:

```
""" Use scikit-build's `cmaker` to control CMake configuration and build.

1. Use `cmaker` to define an object that provides convenient access to
   CMake's configure and build functionality.

2. Use defined object, `maker`, to call `configure()` to read the
   `CMakeLists.txt` file in the current directory and generate a Makefile,
   Visual Studio solution, or whatever is appropriate for your platform.

3. Call `make()` on the object to execute the build with the
   appropriate build tool and perform installation to the local directory.
"""
from skbuild import cmaker
maker = cmaker.CMaker()

maker.configure()

maker.make()
```

2.3 Examples for scikit-build developers

Note: *To be documented.*

Provide small, self-contained setup function calls for (at least) two use cases:

- when a *CMakeLists.txt* file already exists
- when a user wants scikit-build to create a *CMakeLists.txt* file based on the user specifying some input files.

CHAPTER 3

Contributing

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

3.1 Types of Contributions

You can contribute in many ways:

3.1.1 Report Bugs

Report bugs at https://github.com/scikit-build/scikit-build/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.

3.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

3.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

3.1.4 Write Documentation

The scikit-build project could always use more documentation. We welcome help with the official scikit-build docs, in docstrings, or even on blog posts and articles for the web.

3.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/scikit-build/scikit-build/issues.

If you are proposing a new 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 :)

3.2 Get Started

Ready to contribute? Here's how to set up *scikit-build* for local development.

1. Fork the *scikit-build* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/scikit-build.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed (*pip install virtualenvwrapper*), this is how you set up your cloned fork for local development:

```
$ mkvirtualenv scikit-build
$ cd scikit-build/
$ 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 skbuild tests
$ python setup.py test
$ tox
```

   If needed, you can get flake8 and tox by using *pip install* to 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.

3.3 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 2.7, and 3.3, 3.4, 3.5 and PyPy. Check https://travis-ci.org/scikit-build/scikit-build/pull_requests and make sure that the tests pass for all supported Python versions.

3.4 Tips

To run a subset of tests:

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

CHAPTER 4

Credits

Please see the GitHub project page at https://github.com/scikit-build/scikit-build/graphs/contributors

CHAPTER 5

History

PyCMake was created at SciPy 2014 in response to general difficulties building C++ and Fortran based Python extensions across platforms. It was renamed to “scikit-build” in 2016.

How to Make a Release

A core developer should use the following steps to create a release of **scikit-build**.

0. Configure *~/.pypirc* as described here.
1. Make sure that all CI tests are passing.
2. Create the source tarball and binary wheels:

```
git checkout master
git fetch upstream
git reset --hard upstream/master
rm -rf dist/
python setup.py sdist bdist_wheel
```

3. Upload the packages to the testing PyPI instance:

```
twine upload -r pypitest dist/*
```

4. Check the PyPI testing package page.
5. Tag the release. Requires a GPG key with signatures. For version *X.Y.Z*:

```
git tag -s -m "scikit-build X.Y.Z" X.Y.Z upstream/master
```

6. Upload the packages to the PyPI instance:

```
twine upload dist/*
```

7. Check the PyPI package page.
8. Make sure the package can be installed:

```
mkvirtualenv skbuild-pip-install
pip install scikit-build
rmvirtualenv skbuild-pip-install
```

9. Update the version number in *setup.py* and *skbuild/__init__.py* and merge the result.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Resources

- Free software: MIT license
- Documentation: http://scikit-build.readthedocs.io/en/latest/
- Source code: https://github.com/scikit-build/scikit-build
- Mailing list: https://groups.google.com/forum/#!forum/scikit-build