Distutilazy Documentation

Release 0.4.0

Farzad Ghanei

Aug 04, 2018

Contents

Contents:

CHAPTER 1

Command Classes

Command classes are where the functionality of each command is implemented. Multiple command classes can be defined in a single module, for example the *distutilazy.clean* defines:

- *CleanPyc*: Clean compiled files created by CPython (.pyc files and __pycache__ directories)
- *CleanJythonClass*: Clean compiled .class files created by Jython
- *CleanAll*: Clean all temporary file (compiled files, build and dist directories, etc.)

1.1 Using Command Classes

To use command classes, pass the classes (or custom classes extending them) as `cmdclass` argument of `setup()` function in `setup.py` file.

```
import distutilazy.clean.CleanAll
import distutilazy.test.RunTests
from distutils.core import setup

setup(
    cmdclass= {
        "clear": distutilazy.clean.CleanAll,
        "unittests": distutilazy.test.RunTests
    }
)
```

Now to run tests, run this command

```
$ python setup.py unittests
```

To remove all temporary files run

```
$ python setup.py clear
```

Or create a custom class, to extend the ones provided by default:

```
from import distutilazy.clean import CleanAll
from distutils.core import setup

class MyCleaner(CleanAll):
    pass

setup(
    cmdclass= {
        "clear": MyCleaner
    }
)
```

All the command classes extend from `distutils.core.Command` class, and they provide these methods:

initialize_options()
: Initialize options of the command (as attributes of the object). This is called by *distutils.core.Command* after the command object has been constructed.

finalize_options()
: Finalize options of the command (for example to do validation) This is called by *distutils.core.Command* before *run* is called.

run()
: Executes the command with current options state

Here we introduce available modules, and classes they provide.

1.2 distutilazy.clean – Command class to clean temporary files

class distutilazy.clean.**CleanPyc**
: Command class to clean compiled and cached files created by CPython

  root
  : Command option, the path to root directory where cleaning process would affect. (default is current path).

  extensions
  : Command option, a comma separated string of file extensions that will be cleaned

  directories
  : Command option, a comma separated string of directory names that will be cleaned recursively from root path

  default_extensions()
  : Return list of file extensions that are used for compiled Python files

  default_directories()
  : Return list of directory names that are used to store compiled Python files

  find_compiled_files()
  : Return list of absolute paths of all compiled Python files found from the *root* directory recursively.

  find_cache_directories()
  : Return list of absolute paths of all cache directories found from the *root* directory recursively.

class distutilazy.clean.**clean_pyc**
: Alias to *distutilazy.clean.CleanPyc*

class distutilazy.clean.**CleanJythonClass**
: Command class to clean compiled class files created by Jython

root
Command option, the path to root directory where cleaning process would affect. (default is current path).

extensions
Command option, a comma separated string of file extensions that will be cleaned

directories
Command option, a comma separated string of directory names that will be cleaned recursively from root path

default_extensions()
Return list of file extensions that are used for compiled class files

default_directories()
Return list of directory names that are used to store class files

find_class_files()
Return list of absolute paths of all compiled class files found from the *root* directory recursively.

class distutilazy.clean.**CleanAll**
Command class to clean all temporary files (compiled files created by Jython, CPython), build and dist directories, etc.

root
Command option, the path to root directory where cleaning process would affect. (default is current path).

extensions
A command option, a comma separated string of file extensions that will be cleaned

directories
Command option, a comma separated string of directory names that will be cleaned recursively from root path

default_extensions()
Return list of file extensions that are used for file extensions to be cleaned by default.

default_directories()
Return list of directory names that are going to be cleaned by default.

class distutilazy.clean.**clean_all**
Alias to *distutilazy.clean.CleanAll*

1.3 distutilazy.pyinstaller – Command class to run PyInstaller

class distutilazy.pyinstaller.**BdistPyInstaller**
Command class to run PyInstaller

clean
Command option, Boolean value to specify if PyInstaller should clean spec files before bundling the application. (Default is False)

hidden_imports
Command option, a comma separated string of modules to force import (same as –hidden-import option of PyInstaller)

icon
Command option, path to the icon file used for the executable

name
Command option, name of the bundled executable. (Default is found from the current distribution name)

paths
Command option, a list of extra paths to search for modules to import from, separated by the standard path separator of current platform (; on Windows and : on Others).

pyinstaller_path
Command option, path to PyInstaller executable. (By default is "pyinstaller" and will be found on shell path)

target
Command option, path to target file to bundle

windowed
Command option, Boolean value to specify if the application is a windowed application or not. (Default is False)

default_imports()
Return a list of modules to be imported by default

default_paths()
Return a list of extra paths to search for modules by default

default_pyinstaller_opts()
Return a list of PyInstaller options used by default

class distutilazy.pyinstaller.**bdist_pyinstaller**
Alias to *distutilazy.pyinstaller.BdistPyInstaller*

class distutilazy.pyinstaller.**CleanAll**
Command class to clean all temporary files, including PyInstaller spec files. Extends *distutilazy.clean.CleanAll* and has the same attributes and methods.

name
name of the bundled app which is used for the auto generated spec file name.

class distutilazy.pyinstaller.**clean_all**
Alias to *distutilazy.pyinstaller.CleanAll*

1.4 distutilazy.test – Command class to run unit tests

distutilazy.test.**test_suite_for_modules**(*modules*) → unittest.TestSuite
Return a test suite containing test cases found in all the specified modules.

class distutilazy.test.**RunTests**
Command class to find test cases and run them (using standard library unittest)

root
Command option, the path to root directory to find test modules from. If this path is a package and provides __all__, then this list is considered as the list of test modules and no more search happens for other files (Default is "tests").

pattern
Command option, a Unix file name pattern (like fnmatch) to match tests files with. This is used when no files are specified to run, and the *root* is not a package that specifies the tests with its __all__ attr (Default is "test*.py").

files
Command option, a comma separated string of file names to search for test cases. If specified, only the test cases in these files run.

verbosity
Command option, an integer (1 .. 3) specifying the verbosity of the test runner (Default is 1).

get_modules_from_files(*files*)
Accept a list of file paths, import them as modules and return a list of module objects.

find_test_modules_from_package_path(*self*, *package_path*)
Find modules from the package specified by package_path `__all__` attr, import them and return the modules.

find_test_modules_from_test_files(*self*, *root*, *pattern*)
Find files whose name matches the `pattern` from the `root` path, then import them and return the modules.

get_test_runner()
Return a *TestRunner* to run the test suite, configured with the *verbosity* option.

class distutilazy.test.**run_tests**
Alias to *distutilazy.test.RunTests*

CHAPTER 2

Command Package

Distutils allows to specify a package so extra command can be loaded from there. Each command should be defined in a separate module named after the command, which defines a class with the same name.

As a convenient method of using Distutilazy commands, a separate `distutilazy.command` package is provided. For all the command classes available in Distutilazy, a command module exists in this package.

2.1 Using Command Package

To use the commands from this package add `distutilazy.command` package to the list of *command_packages* in `setup.cfg` file.

```
[global]
command_packages = distutilazy.command
```

2.2 Commands

Available modules (hence commands) are:

- bdist_pyinstaller
- clean_all
- clean_jython_classes
- clean_pyc
- test

CHAPTER 3

Introduction

Extra distutils commands, including:

- clean_pyc: clean compiled python files
- clean_jython_class: clean compiled .class files created by Jython
- clean_all: using `distutils.clean`, `clean_pyc` and `clean_jython_class` to clean all temporary files
- bdist_pyinstaller: convenient calls for PyInstaller with sane defaults
- test: run unit tests

An entry point script is also provided to call commands directly, currently only the clean commands are exposed via the script.

CHAPTER 4

Installation

Use `pip` to install from PyPI:

```
$ pip install distutilazy
```

To install from the source, download the source and run

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

There are no specific dependencies, it runs on Python 2.7+ (CPython 2.7, 3.2, 3.3, 3.4 and 3.5, PyPy 2.6 and PyPy3 2.4 are tested). Tests pass on Jython so it should be fine for Jython as well.

An entry point script is also provided to call commands directly, currently only the clean commands are exposed via the script.

CHAPTER 5

How

After installing distutilazy, add `distutilazy.command` package to the list of *command_packages* in your `setup.cfg` file.

```
[global]
command_packages = distutilazy.command
```

That's it. Now you may use new commands directly from your `setup.py`.

To clean compiled python files:

```
$ python setup.py clean_pyc
```

To clean all temporary files (build artifacts, compiled files created by CPython or Jython, etc.):

```
$ python setup.py clean_all
```

Available commands are in `distutilazy.command` package, each command as a separate module.

5.1 Entry Point Script

The *distutilazy* script provides a direct access to the commands. call it with *-h* or *–help* to see available commands. For example this command runs the *clean_all* command (provided by distutilazy package) directly, even without a *setup.py* or *setup.cfg*.

```
$ distutilazy clean_all
```

5.1.1 Development

- Code is hosted on GitHub.
- Documentations are on Read The Docs.

5.2 Tests

If you have make available

```
$ make test
```

You can always use `setup.py` to run tests:

```
$ python setup.py test
```

5.2.1 License

Distutilazy is released under the terms of MIT license.

Python Module Index

d

Index

I

R

T