everyapp.bootstrap

Release 0.2.2

Krys Lawrence

2012-02-22
CONTENTS

1 everyapp.bootstrap 1
  1.1 About . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
  1.2 Features . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
  1.3 Status . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  1.4 Documentation and Support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
  1.5 Contributing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2 Installing and Upgrading 3
  2.1 Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  2.2 Dependencies . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  2.3 About Versioning . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  2.4 Installation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
  2.5 Upgrading . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

3 Usage 7
  3.1 Basic Usage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
  3.2 mkbootstrap . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
  3.3 bootstrap.py . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
  3.4 bootstrap.cfg . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

4 Frequently Asked Questions 19

5 Licence 21
  5.1 Project Licence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
  5.2 Documentation Licence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
  5.3 Licensing of Generated Files . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
  5.4 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

6 Known Bugs 23

7 Related and Alternative Projects 25
  7.1 Related Projects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
  7.2 Alternative Projects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

8 Development 27
  8.1 Development Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
  8.2 Cloning the Repository . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
  8.3 Setting up a Development Environment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
  8.4 Generating the Documentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
  8.5 Packaging a Release . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
# API Documentation

9.1 everyapp (E) ................................................................. 31
9.2 E.bootstrap (E.B) ....................................................... 31
9.3 E.B.customize ........................................................... 31
9.4 E.B.mkbootstrap ......................................................... 32

# Credits

10.1 Authors ................................................................. 33
10.2 Acknowledgements .................................................. 33

# Release Notes

11.1 Release 0.2.2 .......................................................... 35
11.2 Release 0.2.1 .......................................................... 35
11.3 Release 0.2 ............................................................. 35
11.4 Release 0.1.5 .......................................................... 36
11.5 Release 0.1.4 .......................................................... 36
11.6 Release 0.1.3 .......................................................... 36
11.7 Release 0.1.2 .......................................................... 36
11.8 Release 0.1.1 .......................................................... 37
11.9 Release 0.1 ............................................................. 37

# Appendix A: Full Licence Text

12 Appendix A: Full Licence Text ........................................ 39

# Appendix B: Default bootstrap.cfg

13 Appendix B: Default bootstrap.cfg .................................... 51

# Glossary

14 Glossary ................................................................. 55

# Python Module Index

Python Module Index ......................................................... 57

# Index

Index ................................................................. 59
1.1 About

`everyapp.bootstrap` provides an enhanced and customizable `virtualenv` bootstrap script. It aims to make it easy to bootstrap a virtual environment for your project.

This project includes a tool to generate a bootstrap script that will automatically create the virtual environment when run. By placing the bootstrap script in the root of your project’s source tree and making it available on a fresh checkout/clone from your version control repository, you make it easy for anyone who wants to hack on your project (including yourself!) to work in a consistent development environment.

Additionally, this enhanced bootstrap script can read a configuration file and perform additional actions beyond just creating a bare-bones virtual environment.

1.2 Features

- Easy creation of a `virtualenv` bootstrap script.
- Better defaults for `virtualenv`, like using `distribute` instead of `setuptools` and always unzipping EGGs.
- Easy customization of the bootstrapping process.
- Automatic installation of additional distributions using `pip` and/or `easy_install`.
- Easy customization of `pip` and `easy_install` behaviour during distribution installation.
- Automatic execution of additional commands after the bootstrapping is complete.
- Adherence to Semantic Versioning in order to be a well behaved dependency.
1.3 Status

This project is very young and the code should be considered *Alpha* quality. It has been minimally tested on Linux and Windows, under Python 2.6 and 2.7, but has not yet been seriously tested on any platform.

That said, it is largely just a wrapper and a few customizations on the production-grade `virtualenv` project\(^1\), so it should be reasonably stable. It should also be mostly feature complete.

1.4 Documentation and Support

Home page:  http://pypi.python.org/pypi/everyapp.bootstrap

Downloads:  http://pypi.python.org/pypi/everyapp.bootstrap#downloads

Documentation:  See the Table of Contents, or online in the following formats: HTML, EPUB, and PDF.

Bug/issue tracker:  http://bitbucket.org/everyapp/bootstrap/issues

Source code repository:  http://bitbucket.org/everyapp/bootstrap

1.5 Contributing

There is always room for improvement in this project and contributions are certainly welcome. The easiest way to contribute is simply to file a bug report in the issue tracker whenever you find a problem or want to suggest an improvement.

If you would like to participate in a more substantial way, check out the issue tracker, the list of known bugs and the To Do list to find out about the work that still needs to be done.

---

Note:  If you submit a bug report, patch or other code, you automatically agree to licence the contribution. See the licensing information for details on contribution licensing.

---

See also the developer documentation for more information on developing/hacking on `everyapp.bootstrap`.

---

\(^1\) Strictly speaking, `virtualenv` classifies itself as Beta quality, but many consider it to be production-grade.
This chapter describes the requirements, dependencies and steps necessary to install or upgrade *everyapp.bootstrap*.

### 2.1 Requirements

*everyapp.bootstrap* requires the following pre-requisites in order to function:

- A POSIX (Portable Operating System Interface [for Unix]) (Linux, Mac OS X) or Windows system
- Python $\geq 2.4, < 3.0$, or maybe Python 2.3 and the sub-process module

Development systems have the following additional requirements:

- Mercurial to clone the source code repository
- `unix2dos` on POSIX systems to ensure consistent line endings
- `Latex` for PDF (Portable Document Format) documentation generation (optional)

### 2.2 Dependencies

*everyapp.bootstrap* only depends on `distribute` and `virtualenv`. They will be installed automatically.

### 2.3 About Versioning

This project adheres to the Semantic Versioning 1.0.0 specification. This means, among other things, that *everyapp.bootstrap* will be a well-behaved dependency and that increments in the version’s patch number (the third number in the version) should never break API (Application Programming Interface) compatibility.

See Also:

The Semantic Versioning 1.0.0 specification for more information.

### 2.4 Installation

Installing *everyapp.bootstrap* is easy, but there are a few ways to do it. Just follow the instructions in one of the sections below:
2.4.1 Installing With pip

To install using pip, run:

```
pip install everyapp.bootstrap
```

2.4.2 Installing With easy_install

To install using easy_install (i.e. from distribute or setuptools), run:

```
easy_install everyapp.bootstrap
```

2.4.3 Installing From Archive

To install everyapp.bootstrap the old-fashioned way, download the tarball from everyapp.bootstrap on PyPI and then run:

```
tar -zxf everyapp.bootstrap-X.Y.Z.tar.gz
cd everyapp.bootstrap
python setup.py install
```

Note: On Windows you can extract the archive using an appropriate archive tool like 7-Zip, WinZip, or a Windows version of the tar utility.

2.4.4 Installing From Mercurial

To install the latest development version from the Mercurial repository, run:

```
hg clone https://bitbucket.org/everyapp/bootstrap everyapp.bootstrap
cd everyapp.bootstrap
python setup.py install
```

See Also:

Development for additional information on developing/hacking on the development version.

2.5 Upgrading

Upgrading everyapp.bootstrap works very similarly to installing it. Here are some general instructions for performing the upgrade:

2.5.1 Upgrading With pip or easy_install

Upgrading works just like installing (see above), just add the `--upgrade` option before the distribution name. E.g.:

```
pip install --upgrade everyapp.bootstrap
```

or:
easy_install -U everyapp.bootstrap

2.5.2 Upgrading From Archive

Upgrading from the distribution archive works exactly like installing from the archive. Just download the archive of the newer version and follow the installation instruction above.

2.5.3 Upgrading From Mercurial

If you do not already have a clone of the Mercurial repository, then just follow the installation instructions above to perform the upgrade.

If you do already have the repository cloned or kept the one from the original installation, then run the following commands instead:

cd everyapp.bootstrap
hg pull -u
python setup.py install

2.5.4 Final Steps

---

**Note:** Once you have finished the upgrade, do not forget to regenerate your `bootstrap.py` script in your project. See *Usage* for details.

---

**Warning:** The instructions above are the bare minimum necessary to perform an upgrade. Be sure to all review the changes in *Release Notes* for any additional adjustments you may need to perform.
This chapter describes how to use and configure `everyapp.bootstrap`. It starts with some basic usage instructions and then goes on to provide full details on each of the three main components.

### 3.1 Basic Usage

This section provides some basic instructions on how to get started using `everyapp.bootstrap`. It also includes descriptions of some of the new sub-directories that you will find in your project as a result. Finally, some pointers on what to do next are provided.

#### 3.1.1 Getting Started

`everyapp.bootstrap` consists of one main command-line utility, called `mkbootstrap`, that generates a bootstrap script and a configuration file. By default, the bootstrap script is named `bootstrap.py` and the configuration file is named `bootstrap.cfg`.

You use `mkbootstrap` both to generate your first bootstrap files and also to upgrade your `bootstrap.py` when you install newer versions of `everyapp.bootstrap`.

So, to get started, in your project’s root directory, run the command:

```
mkbootstrap
```

This will generate the `bootstrap.py` and `bootstrap.cfg` files.

You can then edit `bootstrap.cfg` as desired, though this is a completely optional step. The `bootstrap.cfg` file lets you customize the behaviour of `bootstrap.py` in several ways like:

- Customizing the default values for `bootstrap.py`’s command-line arguments,
- Specifying additional Python distributions (packages) to install using either `pip` or `easy_install` or both.
- Specifying additional commands to run after the virtual environment has been created and the distributions have been installed.

**See Also:**

`bootstrap.cfg` for information on editing the `bootstrap.cfg` file.

Once you have `bootstrap.cfg` set up the way you want, you are ready to bootstrap the virtualenv environment for your project. To do that, run the command:

```
python bootstrap.py
```
This will create the virtual environment in the current directory (i.e. the root directory for your project).

**Note:** If you want to use a different version of Python than the default one installed on your system, just run `bootstrap.py` using that interpreter. Your virtual environment will then use that version of Python for everything automatically.

The final step is to activate the environment. This is something you should do every time you work on your project. To do it, run the command:

```bash
source ./bin/activate
```

... or, if you are on Windows, run:

```bash
Scripts\activate.bat
```

You are now ready to start hacking on your project using an isolated Python interpreter and only the distributions installed locally in this virtual environment.

**Note:** You will probably want to commit `bootstrap.py` and `bootstrap.cfg` to your project’s version control system (VCS (Version Control System)). That way you can easily generate a consistent development environment each time you clone/check out your project from it’s VCS.

When you activate your environment, your PATH environment variable is altered so that you will run your local version of Python and related commands by default. Also, your shell or command line prompt is altered to remind you that you are in your activated virtual environment.

There is one additional command that is useful from time to time. If you want to deactivate your virtual environment and return to your normal shell or command line, run this command:

```
deactivate
```

See Also:

`mkbootstrap`, `bootstrap.py` and `bootstrap.cfg` for more specific details on each of the three files mentioned above.

See Also:

The virtualenv documentation for more information on virtualenv environments.

### 3.1.2 New Directories

Once you have bootstrapped your virtual environment, you will find several new sub-directories in your project’s root directory. They are briefly described below:

**bin or Scripts** This directory contains the executable programs that are aware of your virtual environment. On Windows the directory is named Scripts, everywhere else it is named bin. In particular you will find the python interpreter, as well as the pip and easy_install commands in this directory.

**lib/pythonX.Y or Lib** This (sub-)directory contains a copy of Python’s standard library, adapted to your virtual environment. You generally won’t touch these files. On Windows the directory is named Lib, everywhere else it is named lib/pythonX.Y, where X and Y are the version numbers of the Python interpreter you are using.

**lib/pythonX.Y/site-packages or Lib/site-packages** This sub-directory contains the virtual environment’s locally installed packages. You can easily install additional packages using pip or easy_install when your virtual environment is activated. On Windows the sub-directory is named Lib/site-packages, everywhere else it is named lib/pythonX.Y/site-packages, where X and Y are the version numbers of the Python interpreter you are using.
On 64-bit POSIX systems (everything except Windows), this symlink is created. It points to the above-mentioned `lib` directory. This is for compatibility.

**Note:** Do not commit these directories to your VCS. In fact, you should probably mark them to be ignored. They will all get re-generated every time you run `bootstrap.py`.

### 3.1.3 Next Steps

Now that you have your project’s development environment set up, you are free continue developing your project as desired. Just make sure to remember these few points:

- Always activate your virtual environment before working on your project.
- Do not forget to update your `bootstrap.cfg` file whenever you install or upgrade a development package.
- Remember that packages installed via `bootstrap.cfg` should only be development support packages. Packages that are actually dependencies of your project should be listed in your project’s `setup.py` file.

Happy hacking!\(^1\)

### 3.2 mkbootstrap

This section provides detailed usage information for the `mkbootstrap` command.

#### 3.2.1 Synopsis

`mkbootstrap [OPTIONS]`

#### 3.2.2 Description

`mkbootstrap` is the command-line utility that generates a `virtualenv` bootstrap script and configuration file for a project. It is also used to upgrade the bootstrap script whenever newer versions of `everyapp.bootstrap` are installed.

#### 3.2.3 Options

`mkbootstrap` accepts several command-line options that are described below. The basic options are the ones that are used most commonly. The advanced options are intended for special situations where more advanced customization is required.

---

\(^1\) This expression is most famously used by Richard M. Stallman.
Basic Options

The following are the most commonly used options of customizing the behaviour of mkbootstrap.

- **version**
  This option tells mkbootstrap to just show program’s version and copyright information, and then exit.

- **h, -help**
  This option tells mkbootstrap to just show program usage information and then exit. It lists all the options described here along with simple descriptions.

- **s FILE_NAME, -script-name FILE_NAME**
  Use this option to give the generated bootstrap script a different name or location. The default is bootstrap.py.

  **Note:** By default, the name of the generated configuration file will also change when this option is given. Use the -c option to override this behaviour.

- **c FILE_NAME, -config-name FILE_NAME**
  Use this option to give the generated configuration file a different name or location. The default is the same as script name but with a .cfg extension.

  **Note:** If you specify a custom location for the configuration file, you will need to always specify it when re-generating/upgrading the bootstrap script.

- **e, -etc**
  Use this option to generate the configuration file in an etc sub-directory instead of in the current directory. This is useful if you want to keep all your project’s configuration files in a common place and not in the project’s root directory. etc is a common convention on POSIX systems.

  **Note:** You can combine this option with the -c option to customize the configuration file name, but then you should not specify a custom location with that option.

- **n, -no-customization**
  This option tells mkbootstrap to only generate a plain vanilla virtualenv bootstrap script, without any customizations or configuration file. The generated bootstrap script is exactly the same as if everyapp.bootstrap had never even been installed.

  This option makes it easy to use everyapp.bootstrap as simply a virtualenv script generator without adding any extra fanciness. This is arguably a bit easier than creating your own bootstrap script generator as described in the virtualenv documentation.

  **Note:** If you use this option, then the available command-line options of the generated bootstrap script will only be the original ones and not include the customized ones provided by this package.

Advanced Options

The options listed here are meant for more advanced customization needs. They allow you to fully customize the generated bootstrap script and configuration file.

- **S FILE_NAME, -customize-script FILE_NAME**
  Normally, mkbootstrap includes it’s own virtualenv customization code in the generated bootstrap script to provide it’s additional functionality. You can use this option to have your own virtualenv customization code
included instead. Just put your customization code in a file and reference it with this option. This is arguably a bit easier than creating your own bootstrap script generator as described in the virtualenv documentation.

Additionally, if you include the $DEFAULT_CONFIG_FILE_NAME$ token in your code, it will be automatically replaced with the name of the configuration file.

See Also:
The virtualenv documentation on creating custom bootstrap scripts for more information on what to put into your customization file.

-C FILE_NAME, -default-config FILE_NAME
Normally, mkbootstrap generates the configuration file from a built-in template. You can use this option, however, to use your own default configuration file instead.

Note: A new configuration file will not be generated if an existing one is already found.

3.2.4 Examples
The simplest and most common usage is to just run it in your project’s root directory. For example:

```bash
cd /path/to/my/project/root
mkbootstrap
```

This will generate a virtualenv bootstrap script called bootstrap.py and a configuration file called bootstrap.cfg in /path/to/my/project/root (i.e. the current directory).

Here are other examples:

```bash
mkbootstrap -s bootdev.py
```

This will generate the files bootdev.py and bootdev.cfg in the current directory.

```bash
mkbootstrap -c venv.cfg
```

This will generate the files bootstrap.py and venv.cfg in the current directory.

```bash
cd /path/to/my/project
mkbootstrap -e
```

This will generate the configuration file as /path/to/my/project/etc/bootstrap.cfg.

3.2.5 Files and Directories

- bootstrap.py The generated enhanced virtualenv bootstrap script
- bootstrap.cfg The configuration file for bootstrap.py

3.2.6 Exit Status

- 0 Successful program execution
- 1 Operational error
- 2 Error parsing command-line options

3.2. mkbootstrap
3.3 bootstrap.py

This section provides detailed usage information for the bootstrap.py command.

3.3.1 Synopsis

python bootstrap.py [OPTIONS]

3.3.2 Description

bootstrap.py is an enhanced virtualenv bootstrap script generated by the mkbootstrap command. The script creates a virtualenv environment, installs additional packages and runs additional commands, as configured, in order to set up a development environment for the project to which it belongs.

Additional packages can be installed using either pip, easy_install or both. The list of packages to install, any customized command-line option defaults and the list of additional post-install commands to run are all configured in a related configuration file, usually called bootstrap.cfg.

3.3.3 Options

bootstrap.py accepts several command-line options that are described below. Most of the options are the same as virtualenv itself or a standard virtualenv bootstrap script, but some have been replaced, modified or added to enhance behaviour.

Note: All the options below can also be set in bootstrap.cfg.

| Warning: | The long versions of the options below should be prefixed by two hyphens (--) not one. The PDF version of this documentation does not render the double hyphens correctly. |

### Standard virtualenv Options

The following options are unchanged from their normal virtualenv behaviour.

- **version**
  This option tells bootstrap.py to just show program’s version and then exit.

- **-h, --help**
  This option tells bootstrap.py to just show program usage information and then exit. It lists all the options described here along with simple descriptions.

- **-v, --verbose**
  Increase the verbosity of program output. This is the opposite of the \(-q\) option. The more times this option is repeated, the higher the verbosity level.

  **Note:** The verbosity level is not currently passed on to pip or easy_install.

- **-q, --quiet**
  Decrease verbosity of the program output. This is the opposite of the \(-v\) option. The more times this option is repeated, the lower the verbosity level.
Note: The verbosity level is not currently passed on to pip or easy_install.

-p PYTHON_EXE, --python PYTHON_EXE
Use this option to specify the Python interpreter to use, e.g., --python=python2.5 will use the python2.5 interpreter to create the new environment. The default is the interpreter used to run bootstrap.py.

-clear
Use this option to clear out the lib directory of any files from previous runs of bootstrap.py.

Note: According to the virtualenv source code, this option does not clear out the bin or any other directories, however, despite what is implied by the message provided by in --help option.

-system-site-packages
By default virtualenv configures the virtual environment to not include access to the system’s global site-packages directory. Use this option to reverse that choice and give access to the global site-packages directory to the virtual environment.

-relocatable
Use this option to make an EXISTING virtualenv environment relocatable. This option fixes up scripts and makes all .pth files, making paths relative.

Note: Do not use this option the first time running bootstrap.py. Call bootstrap.py a second time with this option, to achieve the desired result.

Warning: The authors of everyapp.bootstrap have not tried using this option in the context of bootstrap.py. YMMV.

-extra-search-dir SEARCH_DIRS
This option sets the directory in which to look for setuptools/distribute/pip distributions. You can add any number of additional --extra-search-dir paths. By default the latest distributions are downloaded from PyPI.

-never-download
Force virtualenv to never download anything from the network. Instead, it will fail if local distributions of setuptools/distribute/pip are not present. You can combine this option with the --extra-search-dir to set the location of the local distributions to install.

-prompt PROMPT
Use this option to give an alternative prompt prefix for the virtual environment.

Customized Options

The following options have been added or customized by everyapp.bootstrap to replace some virtualenv options and add additional functionality.

-setuptools
This is the opposite of the virtualenv --distribute option. By default bootstrap.py enables the --distribute option, which tells virtualenv to install distribute instead of setuptools in the virtual environment. Use this option to reverse that choice so that setuptools is installed instead.

-unzip-setuptools
This is the opposite of the virtualenv --unzip-setuptools option. By default bootstrap.py enables the --unzip-setuptools option, which tells virtualenv to unzip the distribute or setuptools EGG in
site-packages instead of leaving it as a compressed EGG file. Use this option to reverse that choice so that distribute or setuptools is left compressed.

-pip-install-options OPTIONS
bootstrap.py can be configured to install additional packages using pip. Use this option to set additional command line options for the pip install command. (Default is ""). Set this to the string of options just as you would give on the command-line.

-easy_install-options OPTIONS
bootstrap.py can be configured to install additional packages using easy_install. Use this option to set additional command line options for the easy_install command. (Default is "-Z", which tells easy_install to always unzip EGG files.) Set this to the string of options just as you would give on the command-line.

3.3.4 Examples

The simplest and most common usage is to just run bootstrap.py after a fresh checkout/clone of a project, in it’s root directory. For example:

```bash
cd /path/to/project/root
python bootstrap.py
```

This will create the virtualenv environment in the project root (i.e. the current directory), install additional packages and run additional commands, as configured, necessary to set up the development environment for the project.

3.3.5 Output

Running bootstrap.py will result in output in several phases. They are:

1. Indication that a configuration file was found.
2. The normal output from virtualenv creating the virtual environment.
3. A list of the packages that will be installed using pip.
4. The normal output of pip installing the packages.
5. A list of the packages that will be installed using easy_install.
6. The normal output of easy_install installing the packages.
7. Indication that post-setup commands will be run.
8. The normal output from running all the commands in sequence.

3.3.6 Files and Directories

- bootstrap.cfg The configuration file for bootstrap.py
- bin or Scripts The directory created by virtualenv for commands and executables.
- lib/pythonX.Y or lib The directory created by virtualenv for Python’s standard library.
- lib/pythonX.Y/site-packages or Lib\site-packages The directory created by virtualenv for the virtual environment’s local packages.
- lib64 A symlink created by virtualenv on 64-bit POSIX systems that points to the above-mentioned lib directory.
3.3.7 Environment Variables

Since bootstrap.py is basically a customized version of virtualenv, it uses all the same environment variables that are used by virtualenv.

See Also:
The virtualenv documentation for details.

3.3.8 Exit Status

0 Successful program execution
1 Operational error
2 Error parsing command-line options
3
   • Attempt to run virtualenv in an (obsolete) workingenv environment, or
   • Using a home directory that contains a space on Windows and not having pywin32 installed, or
   • The Python executable specified with --python does not exist
100 Error running Python interpreter due to it’s sys.prefix value not being what is expected
101 Attempt to run under a Python version less than 2.3
200 Error parsing the configuration file

Note: Other exit codes are possible as they can come from sub-processes like installing distribute, etc.

3.4 bootstrap.cfg

This section provides detailed usage information for the bootstrap.cfg configuration file.

3.4.1 Description

bootstrap.cfg is the configuration file used by the bootstrap.py script to customize it’s behaviour when creating a virtualenv environment for a project. A default bootstrap.cfg is generated, along with the bootstrap.py script, by the mkbootstrap command.

bootstrap.cfg allows one to do the following:
   • Customize the default values of the command-line arguments to bootstrap.py,
   • List additional Python distributions to be installed using pip and/or easy_install, and
   • List additional commands to execute once everything else has been set up.

The bootstrap.cfg file is an ASCII (American Standard Code for Information Interchange) text file, similar in format to INI files in that it is separated into sections in the form of [section_name]. Each section, however, can require a different line format and is not necessarily restricted to a set of key=value pairs.

The full default generated bootstrap.cfg text can be read in Appendix B: Default bootstrap.cfg.
3.4.2 Sections

Each section of the configuration file is described separately below:

[options]

In this section, one can override the default values of the command-line options that bootstrap.py accepts. The accepted options and their default values are listed below.

The format for this section is key = value pairs, one pair per line.

verbose = 0 This option corresponds to the --verbose command-line argument. A higher value means more detailed output.

quiet = 0 This option corresponds to the --quiet command-line argument. A higher value means less detailed output.

python = This option corresponds to the --python command-line argument.

clear = False This option corresponds to the --clear command-line argument. Set it to True to enable the option.

prompt = This option corresponds to the --prompt command-line argument.

system_site_packages = False This option corresponds to the --system-site-packages command-line argument. Set it to True to enable the option.

use_setuptools = False This option corresponds to the --setuptools command-line argument. Set it to True to enable the option.

zip_setuptools = False This option corresponds to the --zip-setuptools command-line argument. Set it to True to enable the option.

pip_install_options = This option corresponds to the --pip-install-options command-line argument.

easy_install_options = -Z This option corresponds to the --easy_install-options command-line argument.

See Also: The bootstrap.py script documentation for more information on the specific purpose of each option.

Note: Hyphens (--) in option names on the command-line are replaced with underscores (_) internally. Use underscores in bootstrap.cfg as well.

[pip]

In this section, one can list any additional Python distributions to install automatically in the virtual environment using pip. These would normally only be the distributions needed for development support, not the dependencies of the project itself (which would go in setup.py).

The format for this section is one distribution per line, each specified either in PEP 314/PEP 345 format using parentheses or in pip-compatible format without parentheses. For example:

pep8 (>=0.5,!=0.5.2,<0.6)

or:

pep8>=0.5,!=0.5.2,<0.6
Note: Not all distributions are compatible with pip. Use the [easy_install] section below for those that are not (or use it for all distributions).

[easy_install]

In this section, like in the [pip] section above, one can list any additional Python distributions to install automatically in the virtual environment. However, distribution listed in this section will be installed using easy_install. Similarly, these would normally only be the distributions needed for development support.

The format for this section is the same as that of the [pip] section above.

[commands]

This section lets one automatically execute additional commands after the virtual environment has been created and all the specified distributions have been installed.

The format for this section is one command (with any arguments) per line.

Note: One can use the $bin$ token to indicate the virtual environment’s bin/ directory, which can have a different name on some platforms (e.g. Scripts\ on Windows). The trailing slash is also included.

For example:

$bin$python setup.py develop

3.4.3 Files and Directories

bootstrap.cfg The default location and name for the configuration file for bootstrap.py, relative to the project’s root directory.

etc/bootstrap.cfg A common alternative location and name for the configuration file for bootstrap.py, relative to the project’s root directory.
FREQUENTLY ASKED QUESTIONS

This chapter lists common questions that are important or have been asked repeatedly, along with their answers.

(None yet.)
CHAPTER
FIVE

LICENCE

This chapter describes the licensing terms for everyapp.bootstrap and its constituent parts. It also provides details on the licence to which contributors automatically agree when contributing copyrightable material to this project.

5.1 Project Licence

everyapp.bootstrap is licensed under the GNU General Public License version 3 or later (GPLv3+). This is free software: you are free to change and redistribute it under certain conditions. There is NO WARRANTY, to the extent permitted by law. For the full licence text, see Appendix A: Full Licence Text.

5.2 Documentation Licence

The documentation source files (those documentation files included in the version control system and/or in the source distribution for this project are considered source code just like the program source code. They are, therefore, also licenced under the GPLv3+ just like the rest of the project.

Additionally, all generated forms of everyapp.bootstrap’s documentation, such as, but not limited to, HTML (Hyper-Text Markup Language), EPUB, latex, PDF, man pages, etc. are also licenced under the GPLv3+, just like the rest of the project.

5.3 Licensing of Generated Files

By default, the generated bootstrap script (typically named bootstrap.py) contains code from this project as well as code from the virtualenv project. When this is the case, it is therefore also licensed under the GPLv3+ as a derivative work. Further, if a project includes this file, but does not depend on it in order to function normally, then the inclusion is considered mere aggregation and does not make the project a derivative work of everyapp.bootstrap.

All copyrights and moral rights on the the bootstrap configuration file that is generated by this distribution (bootstrap.cfg, by default) are explicitly disclaimed. It is in the public domain as it is intended to be copied and modified by anyone using this distribution.

5.4 Contributions

By submitting a patch, bug report or any other copyrightable material to this project, you automatically agree to license the contribution under the Apache Software License, v. 2.0, and further agree that it may be re-licensed as necessary for inclusion in everyapp.bootstrap or other downstream projects.
KNOWN BUGS

Below is a list of known general bugs. There may also be additional bugs listed in the issue tracker.

• Commands and/or paths in bootstrap.cfg that contain spaces, even if quoted, could cause bootstrap.py to blow up. This needs to be tested/ixed for both Linux and Windows.

• Python 3.x is not yet supported.
Chapter 6. Known Bugs
CHAPTER
SEVEN

RELATED AND ALTERNATIVE PROJECTS

This chapter describes other projects that are related in some way to everyapp.bootstrap. It also describes alternative projects that can provide similar functionality.

7.1 Related Projects

Below is a list of other projects that are related to everyapp.bootstrap. If you like this project, you may also find these others useful as well.

**virtualenv:** virtualenv “is a tool to create isolated Python environments.”

everyapp.bootstrap is really just some customizations on top of it.

**Paver:** Paver “is a Python-based build/distribution/deployment scripting tool along the lines of Make or Rake.”

It includes support for creating customized virtualenv bootstrap scripts. In fact, everyapp.bootstrap’s creator, Krys Lawrence, has used Paver very successfully for years. Paver is the inspiration for creating everyapp.bootstrap.

7.2 Alternative Projects

Below is a list of other projects that provide the same or similar functionality. If everyapp.bootstrap does not meet your needs, perhaps one of these other projects will.

**virtualenv:** virtualenv “is a tool to create isolated Python environments.”

While everyapp.bootstrap uses virtualenv to create virtual environment bootstrap scripts, you can use virtualenv directly to create your customized bootstrap scripts. What everyapp.bootstrap adds are the enhanced features like configuration file support and installation of additional packages.

**Paver:** Paver “is a Python-based build/distribution/deployment scripting tool along the lines of Make or Rake.”

It includes support for creating customized virtualenv bootstrap scripts that can install additional packages and run additional commands. However, until recently (2010-11-11) Paver has been only minimally maintained and has not kept up with newer virtualenv developments. Also, Paver is less flexible than everyapp.bootstrap in its virtualenv customizability and its virtualenv support cannot be separated from the rest of the Paver tool. It was for these reasons that everyapp.bootstrap was created. That said, Paver is an excellent tool and the closest alternative in spirit to everyapp.bootstrap.
pyutilib.virtualenv: pyutilib.virtualenv “includes the vpy_create script, which is used to create virtualenv bootstrap scripts that automate the installation of virtualenv along with other Python packages.”

Like everyapp.bootstrap, it includes support for a configuration file and installing additional packages. However, it also seems more complex. It seems to create several additional directories and seems to have a greater focus on Subversion checkouts. everyapp.bootstrap seems to be a simpler, more direct customization of virtualenv.

eenvbuilder: eenvbuilder “is a system for automatically building virtualenvs in Python. To do this, it uses a .env config file to define parcels, which are individual pieces of software to be checked out and installed into the virtualenv.”

eenvbuilder seems like it does similar things as everyapp.bootstrap, however, it is lacking in documentation and like pyutilib.virtualenv, seems to have a greater focus on Subversion checkouts. Also, the concept of parcels and the format of it’s .env files are more complex than the more direct customizations to virtualenv that everyapp.bootstrap provides.

virtualenvwrapper.project: virtualenvwrapper.project “is a plugin [sic] for virtualenvwrapper to extend its project-management features. Where virtualenvwrapper is for managing virtualenv environments, project manages your development source directories.”

Through the use of it’s template system, you could use virtualenvwrapper.project to build a customized development environment for a project. However, virtualenvwrapper does not work on Windows and building/integrating a template into your project as a bootstrap script would likely be more complicated than using everyapp.bootstrap. everyapp.bootstrap is more focused on being part of your project in order to create reproducible development environments. virtualenvwrapper.project seems to have a more external-to-your-project focus.

zc.buildout: The zc.buildout “project provides support for creating applications, especially Python applications. It provides tools for assembling applications from multiple parts, Python or otherwise. An application may actually contain multiple programs, processes, and configuration settings.”

zc.buildout provides an alternative to virtualenv for creating isolated Python environments. Like everyapp.bootstrap it supports using a configuration file and installing additional packages. It can also do a lot more. On the other hand, it does not use virtualenv and is more confusing to use. It is commonly used in automated deployment scenarios, whereas everyapp.bootstrap focuses more on reproducible development environments.

rjm.recipe.venv: rjm.recipe.venv “is a simple zc.buildout recipe that will use the virtualenv package to turn the build-out root into a sandboxed virtual python environment. It is especially useful in conjunction with gp.recipe.pip, which can then be used to install python packages into the virtualenv that has been created.”

rjm.recipe.venv, along with gp.recipe.pip and zc.buildout could be used together to achieve much of the enhanced capabilities of everyapp.bootstrap. However, that is a much more complex setup and seems redundant once you are already using zc.buildout. Also, not all packages can be installed with pip, especially on Windows, which is why everyapp.bootstrap includes support for easy_install.

BPT: BPT “is a Python library (bpt) and a command line application (box) to create and manage isolated environments [sic], or boxes. [...] BPT is similar in some ways to virtualenv, but it is not restricted to Python packages, allowing to install [sic] virtually any Unix software.”

It seems like you could use BPT to create isolated Python environments, like virtualenv does, along with the installation of additional packages and the running of additional commands. However, BPT does not use virtualenv and does not include support for Windows. It also seems more complex because of it’s broader scope. everyapp.bootstrap is simpler and more focused on development environments for Python projects.
The information below might be useful to those that want to hack on the code or contribute in some way.

8.1 Development Overview

Developing everyapp.bootstrap can be great fun. In order to keep things manageable, the source code is stored in a Mercurial repository and a consistent development environment has been built (using everyapp.bootstrap, of course). Additionally, the documentation, in it’s various formats, is all generated from a single set of source files that are readable in plain text form as well.

In order to get started hacking, whether it be on the code or the documentation, you need to clone the repository and bootstrap the development environment. See below for details on how to do both. After that you are free to make your changes and submit patches as desired.

Happy hacking!\(^1\)

8.2 Cloning the Repository

everyapp.bootstrap’s source code is stored in a Mercurial repository on Bitbucket. To check out the latest revision from the repository, run:

```bash
hg clone https://bitbucket.org/everyapp/bootstrap everyapp.bootstrap
```

8.3 Setting up a Development Environment

Once you have cloned the repository (see above), you need to bootstrap the development environment before you can start hacking on the code effectively. To do so, run these commands:

```bash
cd everyapp.bootstrap
python bootstrap.py
source bin/activate
```

On Windows, replace the last line above with:

```bash
Scripts\activate.bat
```

Or, to just install it directly, run:

\(^1\) This expression is most famously used by Richard M. Stallman.
8.4 Generating the Documentation

While the plain text version of the documentation provides most of the relevant information, the full documentation is best experienced in one of the generated formats. The following generated formats are currently supported:

- HTML
- PDF
- EPUB
- man

You can view the generated documentation online, or you can generate the above formats by following the instructions below.

Note: To generate the documentation, you must first set up the development environment (virtualenv environment with all development dependencies already installed). The instructions and code assume that you are working from within one. See above for instructions on how to set one up.

Note: In addition to the tools automatically included in the development environment, you will also need to have Latex installed in order to generate the PDF format. See the Sphinx documentation for details. If you do not have Latex installed, or do not want the PDF version, just omit the latexpdf option or command from the instructions below.

On POSIX systems (Linux, Mac, etc.), run the following commands:

```bash
cd doc/source
make clean html epub latexpdf man
```

On Windows systems, run the following:

```bash
cd doc\source
make clean
make html
make epub
make latexpdf
make man
```

On all platforms the generated documentation will be in the doc/build directory.

8.5 Packaging a Release

This project occasionally produces release distributions. These are stable versions of the code with specific version numbers. Additionally, sometimes it is useful to produce development releases. The instructions below outline how to create release distributions.

To make a release:

1. Update the release date and add important changes in the doc/NEWS.txt file. Commit the changes.
2. Proof read and spell check all source code and documentation. Commit the changes.
3. Tag the release using the version number as the tag. e.g. v0.3.1.
4. Generate the change log with:
   
   ```
   hg log --style changelog > doc/ChangeLog.txt
   ```
5. If running on a POSIX machine (Linux, Mac), fix the line endings with:
   
   ```
   unix2dos doc/ChangeLog.txt
   ```
6. If this is an official release and not a development release, run:
   
   ```
   python setup.py release sdist
   ```
7. Otherwise, run:
   
   ```
   python setup.py sdist
   ```
8. Test the generated source distribution in the `dist/` directory.
9. Upload to PyPI by running:
   
   ```
   python setup.py [release] sdist register upload
   ```
11. Generate the documentation by following *Generating the Documentation*.
12. Copy the PDF and EPUB version into the HTML version by running:
   
   ```
   cd doc/build
   cp latex/everyapp.bootstrap-<version>.pdf html/_static/everyapp.bootstrap.pdf
   cp epub/everyapp.bootstrap-<version>.epub html/_static/everyapp.bootstrap.epub
   ```
   
   Or on Windows, run:
   
   ```
   cd doc\build
   copy latex\everyapp.bootstrap-<version>.pdf html\_static\everyapp.bootstrap.pdf
   copy epub\everyapp.bootstrap-<version>.epub html\_static\everyapp.bootstrap.epub
   ```
13. Upload the documentation by running:
   
   ```
   python setup.py upload_docs
   ```
14. Increase version in `setup.py` for the next release. Make sure this is consistent with Semantic Versioning 1.0.0.
15. Add a new section to `doc/NEWS.txt` for the next release.
16. Commit the changes.
17. Remember to push all changesets to Bitbucket.
18. Update your development environment by running:
   
   ```
   python setup.py develop
   ```

### 8.6 To Do List

Below are some of the tasks that still need to be completed. This is in addition to any bugs listed in *Known Bugs* and/or in the issue tracker.
• Change customize.py so that it emits a warning if no config file is found, rather than saying it has found one. The presence of a config file is the normal/expected case.

• Add advanced options to output the generated files to stdout instead of generating files.

• Add a no-config option that suppresses the generation of the config file in all cases.

• Make customize.py pass on the output verbosity to pip and easy_install when appropriate.

• Create a mailing list? librelist.org?

• Add a link to a zipped archive of the HTML docs.

• Add build scripts. (fabric?)

• Add RPM spec file, DEB config and other extra packaging support.

• See about supporting the singlehtml build in Sphinx (remove unwanted artifacts).

• See about generating an HTML Help project and file on Windows.

• Write an extension that auto-generates autodoc module synopses from module docstrings.

• Write an extension that lets autodoc include classes, functions and data separately so that they can each have their own section headers instead of just “Members”.

• Add terms to the glossary.

• Write tests.

• Add Python 3 support, maybe by using the virtualenv3 or virtualenv5 forks.

• Find and document the way to install LaTeX on Windows in order to generate the PDF documentation on Windows.

### 8.6.1 Documentation To Do List

Below is a list of To Do items related to the proper documentation of everyapp.bootstrap.

---

**Note:** This list is generated automatically. If there are no items below, then the list is currently empty.
CHAPTER NINE

API DOCUMENTATION

This section documents the Application Programming Interface for this project.

Note: For the sake of brevity, the common parent packages of `everyapp` and `everyapp.bootstrap` are sometimes shortened to `E` and `E.B` respectively.

9.1 everyapp (E)

Namespace package for all packages in the `everyapp` collection.

9.2 E.bootstrap (E.B)

Provides enhanced virtualenv environment creation facilities.

9.3 E.B.customize

Customization code for virtualenv bootstrap scripts.

This code is only useful when included in a virtualenv bootstrap script. It uses virtualenv’s supported hooks to enhance and modify the script’s behaviour.

9.3.1 Members

`everyapp.bootstrap.customize.extend_parser(parser)`

virtualenv hook to extend the command-line parser.

This is also where the configuration file is read and overridden default option values are set.

Parameters parser (`optparse.OptionParser`) – The command-line parser to use.

See Also:

`bootstrap.py` for the full list of command-line options, including the new and modified ones.

See Also:

`bootstrap.cfg` for details on the configuration file and overriding default option values.
everyapp.bootstrap.customize.adjust_options (options, args)
    
    virtualenv hook to adjust the parsed command-line options and args.
    
    This function just sets the home directory to the current directory and maps some of the new command-line
    options to their original counterparts.
    
    Parameters
    
    • options (optparse.Values) – The parsed command-line options.
    • args (tuple) – Any positional arguments from the command-line.

    See Also:
    bootstrap.py for the full list of command-line options, including the new and modified ones.

everyapp.bootstrap.customize.after_install (options, home_dir)
    
    virtualenv hook for post-bootstrap customization.
    
    This function is responsible for all the steps that take place after the virtual environment has been created. Specifically, this function installs all the distributions and runs all the commands specified in the configuration file.
    
    Parameters
    
    • options (optparse.Values) – The parsed command-line options.
    • home_dir (str) – The home directory of the virtual environment. (This is always set to ‘.’).

    See Also:
    bootstrap.cfg for details on how to specify what distributions to install and what commands to run.

9.4 E.B.mkbootstrap

Provides the mkbootstrap command.

9.4.1 Members

everyapp.bootstrap.mkbootstrap.main (args=None, program_name=None)
    
    Main program for the mkbootstrap command.
    
    This function will generate the virtualenv bootstrap script in the current directory, and optionally a default configuration file as well.
    
    Parameters
    
    • args (tuple of str, or None) – The command-line arguments to use. If it is None, then sys.args[1:] will be used.
    • program_name (str, or None) – The name of the program as it appears on the command-line. If it is None, then sys.argv[0] is used.

    See Also:
    mkbootstrap for details on the accepted command-line arguments.
This chapter provides the list of the authors of everyapp.bootstrap, as well as a list of acknowledgements to the people and organizations that the author(s) would like to thank.

10.1 Authors

The following people have contributed to the authoring of this program.

**Krys Lawrence <everyapp at krys ca>** *Original author, current maintainer and sole copyright holder*

10.2 Acknowledgements

*I would like to express my sincerest thanks to the people and organizations below (in no particular order), without which this project would not be what it is.*

—Krys Lawrence

**Guido van Rossum <guido at python org>** *Python*

**Ian Bicking <ianb at colorstudy com>** *virtualenv*

**Tarek Ziadé <tarek at ziade org>** *distribute*

**The fellowship of the packaging <distutils-sig at python org>** *distribute*

**Phillip J. Eby <distutils-sig at python org>** *setuptools*

**Georg Brandl <georg at python org>** *Sphinx*

**David Goodger <goodger at python org>** *docutils*

**Sridhar Ratnakumar <srid at nearfar org>** *modern-package-template*

**Richard M. Stallman <rms at gnu org>** *Free software, GNU/Linux, GPL*

**Wingware <http://wingware.com>** *Free licence of WingIDE Professional*

**Van Lindberg <van.lindberg at haynesboone com>** *Licensing terms for patches and bug reports*
Below is the list of the important changes found in each release of everyapp.bootstrap.

### 11.1 Release 0.2.2

*Release date: 2012-02-22*

- Fixed a Windows bug where the virtual environment’s pip was not being used to pip install dependencies.

### 11.2 Release 0.2.1

*Release date: 2012-02-19*

- Fixed a couple bugs in the documentation. No code changes in this release.

### 11.3 Release 0.2

*Release date: 2012-02-19*

- Upgraded dependencies to:
  - distribute 0.6.24
  - virtualenv 1.7.1.2
  - pip 1.1
- Upgraded development dependencies to:
  - Pygments 1.4
  - Jinja2 2.6
  - docutils 0.8.1
  - Sphinx 1.1.2
- Removed these development dependencies:
  - pep8
  - pyflakes
• Added these development dependencies:
  – flake8 1.1
• Removed now-redundant `--global-site-packages` from `bootstrap.py`. Use `--system-site-packages` instead.
• Changed the `bootstrap.cfg` template:
  – `everyapp.bootstrap` 0.2 is now required.
  – Changed the `pip` install examples.
  – `everyapp.bootstrap` is now installed using `pip` instead of `easy_install`.
• `setup.py` will no longer try to replace `setuptools` with `distribute` and so installation using `setuptools` should now be possible.

## 11.4 Release 0.1.5

_Release date: 2010-12-08_

• Fixed a bug with `setup.py` using the wrong `README.txt`.
• Fixed a documentation bug.
• Fixed a bug where the documentation was not being included in the source distribution.

## 11.5 Release 0.1.4

_Release date: 2010-12-08_

• Greatly added and expanded on the documentation.
• PEP 314/PEP 345 dependency listings can now accept an arbitrary amount of whitespace.
• `mkbootstrap -h` now includes links to web resources.
• Default `bootstrap.cfg` file now includes better help comments.
• Fixed a bug where options in `bootstrap.cfg` with more than one equals sign (=) would cause `bootstrap.py` to crash.

## 11.6 Release 0.1.3

_Release date: 2010-09-20_

• Upgraded to `virtualenv` 1.5.1 and removed patch for `virtualenv`’s bug #60.

## 11.7 Release 0.1.2

_Release date: 2010-09-18_

• Documentation improvements and a small change to `mkbootstrap`’s help message.
• Added # prompt = to the [options] section of the default bootstrap.cfg in order to indicate support for virtualenv 1.5's new --prompt option.
• Minimal testing on Linux and Windows under Python 2.6 and 2.7.
• Added an automatic patch for virtualenv's bug #60. This makes bootstrapping work properly under Windows.

11.8 Release 0.1.1

Release date: 2010-09-17

• Upgraded virtualenv dependency to 1.5.
• Fixed an issue with the wrong version of distribute causing the bootstrapping of new projects to fail.

11.9 Release 0.1

Release date: 2010-09-17

• Initial Release.
APPENDIX A: FULL LICENCE TEXT

Below is the full text of the copyright licence for this project, its source code and its documentation.

GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works. By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users. We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors. You can apply it to
your programs, too.

When we speak of free software, we are referring to freedom, not
price. Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights. Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received. You must make sure that they, too, receive
or can get the source code. And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.
For the developers’ and authors’ protection, the GPL clearly explains that there is no warranty for this free software. For both users’ and authors’ sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users’ freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.
An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.


The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its
content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.


No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties’ legal rights to forbid circumvention of technological measures.


You may convey verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

b) The work must carry prominent notices stating that it is released under this License and any conditions added under section
7. This requirement modifies the requirement in section 4 to "keep intact all notices".

c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d) Limiting the use for publicity purposes of names of licensors or authors of the material; or

e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains
a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.


You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible
for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s "contributor version".

A contributor’s "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.


If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs
If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.
Below is the default bootstrap.cfg file that is generated by mkbootstrap.

```plaintext
[options]
## This is the default bootstrap.cfg file for configuring your project’s
## bootstrap.py script. This file is completely optional, but usually useful.

# Increase the default verbosity for the program output. A higher values means
# more detailed output. Use the "quiet" option below to reduce the verbosity.
#verbose = 0

# Decrease the default verbosity for the program output. Higher values means
# less detailed output. Use the "verbose" option above to increase the
# verbosity.
#quiet = 0

# Specify the path to the default Python interpreter to use in the virtualenv
# environment. E.g. "python2.5". By default the interpreter used to run
# bootstrap.py is the one used for the virtual environment.
#python =

# If set to True, the the "lib" directory will be erased every time
# bootstrap.py is run.
#
# Note: According to the virtualenv source code, this option does not clear out
# the "bin" or any other directories, despite what is implied by the message
# provided by in it’s --help option.
#clear = False
```
# Specify an alternate prompt prefix to use when the virtual environment has
# been activated. The default is the normal virtualenv behaviour.
#prompt =

# If set to True, the virtual environment will have access to the system’s
# global "site-packages" directory. Otherwise, the virtual environment will be
# completely isolated from the system’s installed packages.
#system_site_packages = False

# If set to True, setuptools will be installed instead of distribute.
#use_setuptools = False

# If set to True, setuptools or setuptools will not be automatically
# uncompressed and will be left in compressed EGG format.
#zip_setuptools = False

# Default options to pass to the "pip install" command when installing
# additional Python distributions using pip.
#pip_install_options =

# Default options to pass to the "easy_install" command when installing
# additional Python distributions using easy_install.
easy_install_options = -Z

[[pip]]

## In this section, you can list any additional Python distributions to install
## automatically in the virtual environment using pip. These would normally be
## the distributions needed for development, not the dependencies of the
## project itself (which would go in setup.py).
##
## The format for this section is one distribution per line, each specified
## either in PEP314/PEP345 format using parentheses or in pip-compatible format
## without parentheses. For example:
##
## pep8 (>=0.5,!=0.5.2,<0.6)
##
## or:
##
## pep8>=0.5,!=0.5.2,<0.6
##
## Note: Not all distributions are compatible with pip. Use the
## [easy_install] section below for those that are not (or use it for all
## distributions).

everyapp.bootstrap (>=0.2,<0.2.99)
#flake8
#clonedigger (>=1.0.11,<1.1)

[[easy_install]]

## In this section, you can list any additional Python distributions to install
## automatically in the virtual environment using easy_install. These would
## normally be the distributions needed for development, not the dependencies
## of the project itself (which would go in setup.py).
##
## The format for this section is the same as the [pip] section above.
# ipython

[commands]

## This section lets you automatically execute additional commands after the
## virtual environment has been created and all the specified distributions
## have been installed.

## The format for this section is one command (with any arguments) per line.

## Note: You can use the $bin$ token to indicate the virtual environment’s bin/
## directory, which can have a different name on some platforms (e.g. Scripts)\n## on Windows). The trailing slash is also included.

##$bin$python setup.py develop
This chapter provides a glossary of terms used in the everyapp.bootstrap documentation.

(None yet.)
b
everyapp.bootstrap, 31

c
everyapp.bootstrap.customize, 31

e
everyapp, 31

m
everyapp.bootstrap.mkbootstrap, 32
Symbols

–clear
  bootstrap.py command line option, 13
–easy_install-options OPTIONS
  bootstrap.py command line option, 14
–extra-search-dir SEARCH_DIRS
  bootstrap.py command line option, 13
–never-download
  bootstrap.py command line option, 13
–pip-install-options OPTIONS
  bootstrap.py command line option, 14
–prompt PROMPT
  bootstrap.py command line option, 13
–relocatable
  bootstrap.py command line option, 13
–setup tools
  bootstrap.py command line option, 13
–system-site-packages
  bootstrap.py command line option, 13
–version
  bootstrap.py command line option, 12
  mkbootstrap command line option, 10
–zip-setup tools
  bootstrap.py command line option, 13
-C FILE_NAME, –default-config FILE_NAME
  mkbootstrap command line option, 11
-S FILE_NAME, –customize-script FILE_NAME
  mkbootstrap command line option, 10
-c FILE_NAME, –config-name FILE_NAME
  mkbootstrap command line option, 10
-e, –etc
  mkbootstrap command line option, 10
-h, –help
  bootstrap.py command line option, 12
  mkbootstrap command line option, 10
-n, –no-customization
  mkbootstrap command line option, 10
-p PYTHON_EXE, –python PYTHON_EXE
  bootstrap.py command line option, 13
-q, –quiet
  bootstrap.py command line option, 12
-s FILE_NAME, –script-name FILE_NAME
  mkbootstrap command line option, 10
-v, –verbose
  bootstrap.py command line option, 12

A

about, 1
acknowledgements, 33
adjust_options() (in module everyapp.bootstrap.customize), 31
after_install() (in module everyapp.bootstrap.customize), 32
alternative projects, 25
api documentation, 30
everyapp, 31
everyapp.bootstrap, 31
everyapp.bootstrap.customize, 31
everyapp.bootstrap.mkbootstrap, 32
archive
  installation, 4
  upgrading, 5
authors, 33

B

basic usage, 7
directories, 8
getting started, 7
next steps, 9
bootstrap.cfg, 15
default contents, 50
description, 15
directories, 17
files, 17
sections, 15
bootstrap.py, 11
description, 12
directories, 14
environment variables, 14
time limits, 14
exit status, 15
files, 14
options, 12
bootstrap.py command line option
- --clear, 13
- --easy_install-options OPTIONS, 14
- --extra-search-dir SEARCH_DIRS, 13
- --never-download, 13
- --pip-install-options OPTIONS, 14
- --prompt PROMPT, 13
- --relocatable, 13
- --setuptools, 13
- --system-site-packages, 13
- --version, 12
- --zip-setuptools, 13
- -h, --help, 12
- -p PYTHON_EXE, --python PYTHON_EXE, 13
- -q, --quiet, 12
- -v, --verbose, 12
mercurial, 4
pip, 3
release notes, 33
requirements, 3
versioning, 3

K
known bugs, 21

L
licence, 19
  contribution, 21
  documentation, 21
  full text, 37
  generated files, 21
  GPL, 37
  project, 21

M
main() (in module everyapp.bootstrap.mkbootstrap), 32
mercurial
  installation, 4
  upgrading, 5
mkbootstrap, 9
  api, 32
  description, 9
  directories, 11
  examples, 11
  exit status, 11
  files, 11
  options, 9
  synopsis, 9
mkbootstrap command line option
  –version, 10
  -C FILE_NAME, –default-config FILE_NAME, 11
  -S FILE_NAME, –customize-script FILE_NAME, 10
  -c FILE_NAME, –config-name FILE_NAME, 10
  -e, –etc, 10
  -h, –help, 10
  -n, --no-customization, 10
  -s FILE_NAME, –script-name FILE_NAME, 10

N
news, 33
next steps
  basic usage, 9

O
options
  bootstrap.py, 12
  mkbootstrap, 9
other projects
  alternatives, 25
  related, 23
  output
  bootstrap.py, 14

P
packaging a release, 28
pip
  installation, 3
  upgrading, 4
project licence, 21
Python Enhancement Proposals
  PEP 314, 16, 36
  PEP 345, 16, 36

Q
questions, 17

R
related projects, 23
release
  notes, 33
  packaging, 28
  versioning, 3
requirements, 3

S
sections
  bootstrap.cfg, 15
status, 1
support, 2
synopsis
  bootstrap.py, 12
  mkbootstrap, 9

T
thanks, 33
to do list, 29
documentation, 30

U
upgrading, 4
  archive, 5
  dependencies, 3
  easy_install, 4
  final steps, 5
  mercurial, 5
  pip, 4
  release notes, 33
  requirements, 3
  versioning, 3
usage, 5
  basic usage, 7

Index 61
Index

everyapp.bootstrap, Release 0.2.2

- bootstrap.cfg, 15
- bootstrap.py, 11
- default bootstrap.cfg, 50
- directories, 8
- getting started, 7
- mkbootstrap, 9
- next steps, 9

V

versioning, 3