
Welcome to version.py's documentation!

Release 0.2.6

Jul 16, 2019

Contents

1	Installation	3
1.1	Writing your module's <code>version.py</code> :	3
1.2	Configuring <code>git</code> to auto-version:	3
1.3	Configuring your <code>setup.py</code> to use <code>version.py</code> :	3
2	Contributing	5

Version: 0.2.6

Documentation: <https://versionpy.readthedocs.io>

Source: <https://github.com/gregschmit/version.py>

PyPI: <https://pypi.org/project/version-py/>

The Problem: I don't like manually incrementing versions very often, and I want a hook in my program for people to get the correct version whether they have the `git` source or if they install it from PYPI.

The Solution: If you include this script in your package and perform some relatively minor adjustments to your `setup.py`, every time you build `sdist/bdist_wheel`, the version will be "stamped" into your distribution, so users will know from the software itself which version they have. If your users downloaded your software with `git`, then the `version.py` will get the version from `git`.


```
pip install version-py
```

1.1 Writing your module's `version.py`:

```
python3 -m version_py -o /path/to/repo/package
```

For almost all cases, this file should be **one directory below your “`setup.py`”**. This file should be a part of your actual package, and typically your `setup.py` is one directory above your package.

1.2 Configuring `git` to auto-version:

To start versioning at 0.1.x, just do `git tag -a v0.1 -m 'version 0.1'`. Each commit will increment the PATCH level by 1 according to this module's `get_version()`. When you want to increment the MINOR, just do `git tag -a v0.2 -m 'version 0.2'`.

1.3 Configuring your `setup.py` to use `version.py`:

In your `setup.py` you should stamp the directory so when you generate your package sdist/bdist, the `VERSION_STAMP` exists (since your sdist/bdist won't have the `.git` directory telling the user which version they have):

```
from my_module_name import version

version.stamp_directory('./my_module_name')
```

You should also get the version from `get_version` and ensure your `VERSION_STAMP` gets included in your sdists/bdists, like:

```
...
setup(
    name='my-module-name',
    version=version.get_version(),
    packages=find_packages(),
    include_package_data=True,
    package_data={'my_module_name': ['VERSION_STAMP']},
    ...
)
```

At the end of your `setup.py` you should un-stamp the directory:

```
...
version.unstamp_directory('./my_module_name')
```

Finally, source distributions don't listen to `package_data` instructions, so you need to include the following in a `MANIFEST.in` file in the same directory as `setup.py`:

```
include */VERSION_STAMP
```


CHAPTER 2

Contributing

Email gschmi4@uic.edu if you want to contribute. You must only contribute code that you have authored or otherwise hold the copyright to, and you must make any contributions to this project available under the MIT license.

To collaborators: don't push using the `--force` option.