Bump'R Documentation

Release 0.1.0

Axel Haustant

August 22, 2013

CONTENTS

Bump'R is a version bumper and releaser allowing in a single command:

- Clean-up release artifact
- Bump version and tag it
- Build a source distribution and upload on PyPI
- Update version for a new development cycle

Bump'R intend to be customizable with the following features:

- Optionnal test suite run before bump
- Customizable with a config file
- Overridable by command line
- Extensible with hooks

CHAPTER

ONE

COMPATIBILITY

Bump'R requires Python 2.6+

CHAPTER

TWO

INSTALLATION

You can install Bump'R with pip:

```
$ pip install bumpr
```

or with easy_install:

```
$ easy_install bumpr
```

CHAPTER

THREE

USAGE

You can use directly the command line to setup every parameter:

```
$ bumpr fake/__init__.py README.rst -M -ps dev
```

But Bump'R is designed to work with a configuration file (`bumpr.rc` by defaults). Some features are only availables with the configuration file like:

- commit message customization
- hooks configuration
- multiline test, clean and publish commands

Here's an exemple:

```
[bumpr]
file = fake/__init__.py
vcs = git
tests = tox
publish = python setup.py sdist register upload
clean =
    python setup.py clean
    rm -rf *egg-info build dist
files = README.rst

[bump]
unsuffix = true
message = Bump version {version}

[prepare]
suffix = dev
message = Prepare version {version} for next development cycle

[changelog]
file = CHANGELOG.rst
bump = {version} ({date:%Y-%m-%d})
prepare = In development

[readthedoc]
id = fake
```

This way you only have to specify which part you want to bump on the command line:

```
$ bumpr -M  # Bump the major
$ bumpr     # Bump the default part aka. patch
```

CHAPTER

FOUR

DOCUMENTATION

4.1 Configuration file

The `bumpr.rc` configuration file is an inifile with the following possible sections and keys.

4.1.1 bumpr

This is the main section defining the common behavior and parameters.

file *Default:* `None`

The file containing the version string to extract.

regex *Default:* `r'(__version__|VERSION)\s*=\s*(\'|")(?P<version>.+?)(\'|")'`

The regex used to extract the version string. It must have a `version` named group.

encoding *Default:* `utf8`

The files encoding.

vcs *Default:* `None`

commit *Default:* `True`

If `True` and vcs is defined, commit the changes.

tag: *Default:* `True`

If `True` and vcs is defined, tag the version.

verbose *Default:* `False`

If `True`, display verbose output and command line output.

dryrun *Default:* `False`

TODO

clean *Default:* `None`

Specify the commands to be executed on the *clean* phase. Should have a single command by line.

tests *Default:* `None`

Specify the commands to be executed on the *test* phase. Should have a single command by line.

publish *Default:* `None`

Specify the commands to be executed on the *publish* phase. Should have a single command by line.

files *Default:* `[]`

Extra files to process. Those files will be processed by hooks to. Specify one file by line.

4.1.2 bump

This section define the bump phase behavior.

unsuffix *Default:* `True`

If `True` the current verion suffix will be removed.

suffix: *Default:* `None`

If set, this suffix will ba appended to the version.

part: *Default:* `None`

Specify the part to bump between `major`, `minor` or `patch`.

message *Default:* `Bump version {version}`

Specify the commit message that will be bumped. You can use the following token in your format pattern: `version`, `major`, `minor`, `patch` and `date`. All formating operations are accepted.

4.1.3 prepare

This section define the prepare phase behavior.

unsuffix *Default:* `False`

If `True` the current verion suffix will be removed.

suffix: *Default:* `None`

If set, this suffix will ba appended to the version.

part: *Default:* `patch`

Specify the part to bump between `major`, `minor` or `patch`.

message *Default:* `Update to version {version} for next development cycle`

Specify the commit message that will be bumped. You can use the following token in your format pattern: `version`, `major`, `minor`, `patch` and `date`. All formating operations are accepted.

4.1.4 hooks

Each hook can contribute to configuration with its own section.

See *Hooks*.

4.1.5 sample

Here a sample `bumpr.rc` file

```
[bumpr]
file = fake/__init__.py
vcs = git
tests = tox
publish = python setup.py register sdist upload
clean =
    python setup.py clean
    rm -rf *egg-info build dist
files = README.rst

[bump]
message = 'Commit version {version}'

[prepare]
suffix = dev
message = Prepare version {version} for next development cycle

[changelog]
file = CHANGELOG.rst
bump = {version} ({date:%Y-%m-%d})
prepare = In development

[readthedoc]
id = bumpr
```

4.2 Command line usage

```
$ bumpr -h
usage: bumpr [-h] [--version] [-M] [-m] [-p] [-s SUFFIX] [-u] [-pM] [-pm]
             [-pp] [-ps PREPARE_SUFFIX] [-pu] [--vcs {git,hg}] [-v]
             [-c CONFIG] [-d]
             [file] [files [files ...]]

Version bumper and Python package releaser

positional arguments:
  file                  Versionned module file
  files                 Files to update

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit
  -M, --major           Bump major version
  -m, --minor           Bump minor version
  -p, --patch           Bump patch version
  -s SUFFIX, --suffix SUFFIX
                        Set suffix
  -u, --unsuffix        Unset suffix
  -pM, --prepare-major  Bump major version
  -pm, --prepare-minor  Bump minor version
  -pp, --prepare-patch  Bump patch version
  -ps PREPARE_SUFFIX, --prepare-suffix PREPARE_SUFFIX
                        Set suffix
  -pu, --prepare-unsuffix
                        Unset suffix
  --vcs {git,hg}        VCS implementation
```

```
-v, --verbose         Verbose output
-c CONFIG, --config CONFIG
                      Specify a configuration file
-d, --dryrun          Do not write anything and display a diff
```

4.3 Hooks

4.3.1 Read the doc

4.3.2 Changelog

4.3.3 Commands

4.4 Changelog

4.4.1 0.1.0 (2013-08-22)

- Initial release. Missing some parts but working!

CHAPTER

FIVE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*