
dover Documentation

Release 0.3.0

Mark Gemmill

Mar 03, 2018

Contents:

1	Installation	3
2	What does it do?	5
3	Usage	7
4	Basics	9
4.1	Fetch Current Project Version	9
4.2	Currently Tracked File Status	9
4.3	Reviewing Version Increment Changes	10
4.4	Applying Version Increment Changes	10
5	Examples	11
6	What If There Is a Problem?	13
7	Indices and tables	15

A commandline utility for incrementing your project version numbers.

CHAPTER 1

Installation

```
... pip install dover
```


CHAPTER 2

What does it do?

When `dover` is run from the root directory of your project, it does the following:

1. looks for a configuration file (`.dover`, `.dover.ini`, `setup.cfg`)
2. reads any `dover` configuration line in this format:

```
[dover:file:relatvie/file.pth]
```

3. searches the configured file references for “version” strings
4. validates all version strings across all configured files.
5. displays and/or increments the version strings based upon cli options.

CHAPTER 3

Usage

```
... dover --help

dover v0.3.0

dover is a commandline utility for
tracking and incrementing your
project version numbers.

Usage:
  dover          [--list] [--debug]
  dover increment ((--major|--minor|--patch) [--alpha|--beta|--rc] |
                  [--major|--minor|--patch] (--alpha|--beta|--rc))
                  [--apply] [--debug] [--no-list]

Options:
  -M --major    Update major version segment.
  -m --minor    Update minor version segment.
  -p --patch    Update patch version segment.
  -a --alpha    Update alpha pre-release segment.
  -b --beta     Update beta pre-release segment.
  -r --rc       Update release candidate segment.
  --debug       Print full exception info.
  -h --help     Display this help message
  --version     Display dover version.
```


`dover` will look for either a specific **.dover** configuration file or a **setup.cfg** file within the current directory.

Files that are to be tracked by `dover` should be listed in the following format:

```
[dover:file:relative-file-path.ext]
```

Example:

```
[dover:file:setup.py]
[dover:file:setup.cfg]
[dover:file:dover/cli.py]
```

4.1 Fetch Current Project Version

Calling `dover` without any arguments returns the **current** version number of the project.

```
... dover
0.1.0
```

This is useful for capturing your project version for use with other commandline utilities.

4.2 Currently Tracked File Status

Calling `dover` with the `--list` option, prints the project version and the list of all files and version strings being tracked, and the line they appear on:

```
... dover --list
Current Version: 0.1.0
Files:
  setup.py      0005 (__version__ = '0.1.0')
```

```
setup.cfg      0002 (version = 0.1.0)
doover/cli.py  0025 (__version__ = '0.1.0')
```

4.3 Reviewing Version Increment Changes

Calling `doover increment` with one the the segment options (e.g. `--minor`), will print a listing of the propsed version change and the files that will be effected:

```
... doover increment --minor
Current Version: 0.1.0
New Version:     0.2.0
Files:
  setup.py       (0.1.0 -> 0.2.0)
  setup.cfg      (0.1.0 -> 0.2.0)
  doover/cli.py  (0.1.0 -> 0.2.0)
```

Attention: Only the use of the `--apply` option will perform a update to your files.

4.4 Applying Version Increment Changes

To save the change make the same call with the `--apply` option:

```
... doover increment --minor --apply
Current Version: 0.1.0
New Version:     0.2.0
Files:
  setup.py       (0.1.0 -> 0.2.0)
  setup.cfg      (0.1.0 -> 0.2.0)
  doover/cli.py  (0.1.0 -> 0.2.0)
Version updates applied.
```

CHAPTER 5

Examples

Applying a pre-release option (`-alpha`, `-beta` or `-rc`), will just append the pre-release to the current version:

```
... dover increment --alpha --no-list
Current Version: 0.1.0
New Version:    0.1.0-alpha
```

Applying a pre-release option to an existing pre-release of the same name increments the pre-release:

```
... dover increment --alpha --no-list
Current Version: 0.1.0
New Version:    0.1.0-alpha.1
```

Applying a pre-release option with a segment option, increments the segment and appends the pre-release value:

```
... dover increment --minor --alpha --no-list
Current Version: 0.1.0
New Version:    0.2.0-alpha
```

What If There Is a Problem?

If at any point the version numbers between files being tracked are missaligned, `dover` will raise an error:

```
... dover increment --major --apply
Not all file versions match:

setup.py      0.1.0  (__version__ = '0.1.0')
setup.cfg     0.3.0  (version = 0.3.0)
dover/cli.py  0.1.0  (__version__ = '0.1.0')
```


CHAPTER 7

Indices and tables

- search