
zest.releaser Documentation

Release

March 18, 2013

CONTENTS

Make easy, quick and neat releases of your Python packages. You need to change the version number, add a new heading in your changelog, record the release date, svn/git/bzr/hg tag your project, perhaps upload it to pypi... *zest.releaser* takes care of the boring bits for you.

Here's an overview of the documentation we have for you. First the documentation on *using* *zest.releaser*:

PACKAGE RELEASING MADE EASY: ZEST.RELEASER OVERVIEW AND INSTALLATION

`zest.releaser` is collection of command-line programs to help you automate the task of releasing a Python project.

It does away with all the boring bits. This is what `zest.releaser` automates for you:

- It updates the version number. The version number can either be in `setup.py` or `version.txt`. For example, it switches you from `0.3.dev0` (current development version) to `0.3` (release) to `0.4.dev0` (new development version).
- It updates the history/changes file. It logs the release date on release and adds a new heading for the upcoming changes (new development version).
- It tags the release. It creates a tag in your version control system named after the released version number.
- It optionally uploads a source release to PyPI. It will only do this if the package is already registered there (else it will ask, defaulting to 'no'); `zest releaser` is careful not to publish your private projects!

1.1 Most important URLs

First the three most important links:

- The full documentation is at zestreleaser.readthedocs.org.
- We're on [PyPI](https://pypi.org), so we're only an `pip install zest.releaser` away from installation on your computer.
- The code is at github.com/zestsoftware/zest.releaser.

And... we're automatically being tested by Travis:

1.2 Installation

Just a simple `pip install zest.releaser` or `easy_install zest.releaser` is enough.

Alternatively, buildout users can install zest.releaser as part of a specific project's buildout, by having a buildout configuration such as:

```
[buildout]
parts =
    scripts

[scripts]
recipe = zc.recipe.egg
eggs = zest.releaser
```

1.3 Version control systems: svn, hg, git, bzt

Of course you must have a version control system installed. zest.releaser currently supports:

- Subversion (svn).
- Mercurial (hg).
- Git (git).
- Bazaar (bzt).

Others could be added if there are volunteers! Git and mercurial support have been contributed years ago when we were working with bzt and subversion, for instance.

1.4 Available commands

Zest.releaser gives you four commands to help in releasing python packages. They must be run in a version controlled checkout. The commands are:

- **prerelease**: asks you for a version number (defaults to the current version minus a 'dev' or so), updates the setup.py or version.txt and the CHANGES/HISTORY/CHANGELOG file (with either .rst/.txt/.markdown or no extension) with this new version number and offers to commit those changes to subversion (or bzt or hg or git)
- **release**: copies the the trunk or branch of the current checkout and creates a version control tag of it. Makes a checkout of the tag in a temporary directory. Offers to register and upload a source dist of this package to PyPI (Python Package Index). Note: if the package has not been registered yet, it will not do that for you. You must register the package manually (`python setup.py register`) so this remains a conscious decision. The main reason is that you want to avoid having to say: "Oops, I uploaded our client code to the internet; and this is the initial version with the plaintext root passwords."
- **postrelease**: asks you for a version number (gives a sane default), adds a development marker to it, updates the setup.py or version.txt and the CHANGES/HISTORY/CHANGELOG file with this and offers to commit those changes to version control. Note that with git and hg, you'd also be asked to push your changes (since 3.27). Otherwise the release and tag only live in your local hg/git repository and not on the server.
- **fullrelease**: all of the above in order.

There are two additional tools:

- **longtest**: small tool that renders a `setup.py`'s long description and opens it in a web browser. This assumes an installed `docutils` (as it needs `rst2html.py`).
- **lasttagdiff**: small tool that shows the diff of the currently committed trunk with the last released tag. Handy for checking whether all the changes are adequately described in the changes file.

UPLOADING TO PYPI (OR CUSTOM SERVERS)

When the (full)release command tries to upload your package to a pypi server, zest.releaser basically just executes the command `python setup.py register sdist --formats=zip upload`.

For safety reasons zest.releaser will *only* offer to upload your package to <http://pypi.python.org> when the package is already registered there. If this is not the case yet, you get a confirmation question whether you want to create a new package.

If the `setup.py register ...` command fails, you probably need to configure your PyPI configuration file. And of course you need to have `setuptools/distribute` installed.

2.1 PyPI configuration file (~/.pypirc)

For uploads to PyPI to work you will need a `.pypirc` file in your home directory that has your pypi login credentials, like this:

```
[server-login]
username:maurits
password:secret
```

2.2 Uploading to other servers

Since python 2.6 (or in earlier python versions, with `collective.dist`) you can specify multiple indexes for uploading your package in your `.pypirc`:

```
[distutils]
index-servers =
    pypi
    local

[pypi]
#pypi.python.org
username:maurits
password:secret

[local]
repository:http://localhost:8080/test/products/
username:maurits
```

```
password:secret
# You may need to specify the realm, which is the domain the
# server sends back when you do a challenge:
#realm:Zope
```

See <http://pypi.python.org/pypi/collective.dist> for more info.

When all this is configured correctly, `zest.releaser` will first register and upload at the official pypi (if the package is registered there already). Then it will offer to upload to the other index servers that you have specified in `.pypirc`.

Note that since version 3.15, `zest.releaser` also looks for this information in the `setup.cfg` if your package has that file. One way to use this, is to restrict the servers that `zest.releaser` will ask you upload to. If you have defined 40 index-servers in your `pypirc` but you have the following in your `setup.cfg`, you will not be asked to upload to any server:

```
[distutils]
index-servers =
```

Note that after creating the tag we still ask you if you want to checkout that tag for tweaks or pypi/distutils server upload. We could add some extra checks to see if that is really needed, but someone who does not have index-servers listed, may still want to use an entry point like `gocept.zestreleaser.customupload` to do uploading, or do some manual steps first before uploading.

Some people will hardly ever want to do a release on PyPI but in 99 out of 100 cases only want to create a tag. They won't like the default answer of 'yes' to that question of whether to create a checkout of the tag. So since version 3.16 you can influence this default answer. You can add some lines to the `.pypirc` file in your home directory to change the default answer for all packages, or change it for individual packages in their `setup.cfg` file. The lines are this:

```
[zest.releaser]
release = no
```

You can use no/false/off/0 or yes/true/on/1 as answers; upper, lower or mixed case are all fine.

2.3 Including all files in your release

By default, only the Python files and a `README.txt` are included (by `setuptools`) when you make a release. So you miss out on your changelog, json files, stylesheets and so on. There are two strategies to include those other files:

- Add a `MANIFEST.in` file in the same directory as your `setup.py` that lists the files you want to include. Don't worry, wildcards are allowed. Actually, `zest.releaser` will suggest a sample `MANIFEST.in` for you if you don't already have it. The default is often good enough.
- `Setuptools` *can* detect which files are included in your version control system (svn, git, etc.) which it'll then automatically include.

The last approach has a problem: not every version control system is supported out of the box. So you might need to install extra packages to get it to work. So: use a `MANIFEST.in` file to spare you the trouble. If not, here are some extra packages:

- `setuptools-git` (Setuptools plugin for finding files under Git version control)
- `setuptools_hg` (Setuptools plugin for finding files under Mercurial version control)

- `setuptools_bzr` (Setuptools plugin for finding files under Bazaar version control)
- `collective.dist` (when using python2.4, depending on your `~/.pypirc` file)
- `setuptools_subversion` (Setuptools plugin for finding files under Subversion version control.) You probably need this when you upgrade to the recent subversion 1.7. If you suddenly start missing files in the sdist's you upload to PyPI you definitely need it. Alternatively: set up a proper `MANIFEST.in` as that method works with any version control system.

In general, if you are missing files in the uploaded package, the best is to put a proper `MANIFEST.in` file next to your `setup.py`. See [zest.pocompile](#) for an example.

2.4 Running automatically without input

Sometimes you want to run `zest.releaser` without hitting `<enter>` all the time. You might want to run `zest.releaser` from your automatic test environment, for instance. For that, there's the `--no-input` commandline option. Pass that and all defaults will be accepted automatically.

This means your version number and so must be OK. If you want to have a different version number from the one in your `setup.py`, you'll need to change it yourself by hand. And the next version number will be chosen automatically, too. So `1.2` will become `1.3`. This won't detect that you might want to do a `1.3` after a `1.2.1` bugfix release, but we cannot perform feats of magic in `zest.releaser` :-)

In case you always want to accept the defaults, a setting in your `setup.cfg` is available:

```
[zest.releaser]
no-input = true
```

An important reminder: if you want to make sure you never upload anything automatically to the python package index, include the `release = no` setting in `setup.cfg`:

```
[zest.releaser]
no-input = true
release = no
```


ASSUMPTIONS

Zest.releaser originated at [Zest software](#) so there are some assumptions build-in that might or might not fit you. Lots of people are using it in various companies and open source projects, so it'll probably fit :-)

- If you are using svn, your svn is structured with /trunk, /tags (or /tag) and optionally /branches (or /branch). Both a /trunk or a /branches/something checkout is ok.
- There's a version.txt or setup.py in your project. The version.txt has a single line with the version number (newline optional). The setup.py should have a single `version = '0.3'` line somewhere. You can also have it in the actual `setup()` call, on its own line still, as `“ version = '0.3',“`. Indentation and the comma are preserved. If you need something special, you can always do a `version=version` and put the actual version statement in a zest.releaser-friendly format near the top of the file. Reading (in Plone products) a version.txt into setup.py works great, too.
- The history/changes file restriction is probably the most severe at the moment. `zest.releaser` searches for a restructuredtext header with parenthesis. So something like:

```
Changelog for xyz
=====

0.3 (unreleased)
-----

- Did something

0.2 (1972-12-25)
-----

- Reinout was born.
```

That's just the style we started with. Pretty clear and useful.

And documentation on `zest.releaser` as a project; for instance for reporting bugs and fixing the code:

IMPROVING ZEST.RELEASER: REPORT BUGS, FORK ON GITHUB OR EMAIL US

Did you find a bug? Do you have an improvement? Do you have questions? We run zest.releaser as a proper open source project on github at <https://github.com/zestsoftware/zest.releaser>, so you have three basic options:

- Feel free to report bugs on [our github issue tracker](#). And feature requests, too. Normally you'll get a quick reply within a day or so, depending on our relative timezones. If you don't get an answer within a few days, please send off a quick email to remind us.
- Hey, we're on github, so fork away to your heart's delight. We got a couple of nice fixes and additions in this way.

If you are going to fork zest.releaser, take a look at [Information for developers of zest.releaser](#) for setup and test running information.

- Email Reinout and Maurits at reinout@vanrees.org and maurits@vanrees.org. Please email us both at the same time, this way at least one of us can give you a quick reply.

CREDITS

- [Reinout van Rees](#) (Nelen & Schuurmans) is the originator and main author.
- [Maurits van Rees](#) (Zest Software) added a heapload of improvements.
- [Kevin Teague](#) (Canada's Michael Smith Genome Sciences Center) added support for multiple version control systems, most notable Mercurial.
- [Wouter vanden Hove](#) (University of Gent) added support for uploading to multiple servers, using `collective.dist`.
- [Godefroid Chapelle](#) (BubbleNet) added `/tag` besides `/tags` for subversion.

INFORMATION FOR DEVELOPERS OF ZEST.RELEASER

6.1 Running tests

We like to use `zc.buildout` to get a test environment with any versions of needed python packages pinned. When you are in the root folder of your `zest.releaser` checkout, do this:

```
$ python bootstrap.py # Or a different python version.  
$ bin/buildout  
$ bin/test
```

6.2 Python versions

The tests currently pass on python 2.4 (probably), 2.5 and 2.6 and 2.7. Using 2.6.2 specifically will give test failures. Travis continuous integration tests 2.5, 2.6 and 2.7 for us automatically.

6.3 Necessary programs

To run the tests, you need to have the supported versioning systems installed: `svn`, `hg` (mercurial), `git` and `git-svn`. On ubuntu:

```
$ sudo apt-get install subversion git bzr mercurial git-svn
```

There may be test failures when you have different versions of these programs. In that case, please investigate as these may be genuine errors.

6.4 Setuptools is needed

You also need `distribute` (or `setuptools`) in your system path. This is because we basically call `'sys.executable setup.py egg_info'` directly (in the tests and in the actual code), which will give an import error on `setuptools` otherwise. There is a branch with a hack that solves this but it sounds too hacky.

6.5 Configuration for mercurial, bazaar and git

For mercurial you may get test failures because of extra output like this:

```
No username found, using 'name@domain' instead.
```

To avoid this, create a file `~/.hgrc` with something like this in it:

```
[ui]
username = Author Name <email@address.domain>
```

If you keep having problems, [Ubuntu explains it](#) quite good.

There is a similar problem with bazaar. Since bazaar version 2.2, it refuses to guess what your username and email is, so you have to set it once, like this, otherwise you get test failures:

```
$ bzr whoami "Author Name <email@address.domain>"
```

And the same for git, so you should do:

```
$ git config --global user.name "Your Name"
$ git config --global user.email you@example.com
```

For release testing we expected a functioning `.pyirc` file in your home dir, with an old-style configuration, but the tests now use an own config file.

ENTRYPOINTS: EXTENDING/CHANGING ZEST.RELEASER

A `zest.releaser` entrypoint gets passed a data dictionary and that's about it. You can do tasks like generating documentation. Or downloading external files you don't want to store in your repository but that you do want to have included in your egg.

Every release step (prerelease, release and postrelease) has three points where you can hook in an entry point:

before Only the `workingdir` and `name` are available in the data dictionary, nothing has happened yet.

middle All data dictionary items are available and some questions (like new version number) have been asked. No filesystem changes have been made yet.

after The action has happened, everything has been written to disk or uploaded to pypi or whatever.

For the release step it made sense to create one extra entry point:

after_checkout The middle entry point has been handled, the tag has been made, a checkout of that tag has been made and we are now in that checkout directory. Of course, when the user chooses not to do a checkout, this entry point never triggers.

Note that an entry point can be specific for one package (usually the package that you are now releasing) or generic for all packages. An example of a generic one is [gocept.zestreleaser.customupload](#), which offers to upload the generated distribution to a chosen destination (like a server for internal company use). If your entry point is specific for the current package only, you should add an extra check to make sure it is not run while releasing other packages; something like this should do the trick:

```
def my_entry_point(data):
    if data['name'] != 'my.package':
        return
    ...
```

7.1 Entry point specification

An entry point is configured like this in your `setup.py`:

```
entry_points={
    #'console_scripts': [
        # 'myscript = my.package.scripts:main'],
    'zest.releaser.prereleaser.middle': [
        'dosomething = my.package.some:some_entrypoint',
    ]},
```

Replace `prereleaser` and `middle` in `zest.releaser.prereleaser.middle` with `prerelease/release/postrelease` and `before/middle/after` where needed.

See the `setup.py` of `zest.releaser` itself for some real world examples.

You'll have to make sure that the `zest.releaser` scripts know about your entry points, for instance by placing your egg (with entry point) in the same `zc.recipe.egg` section in your buildout as where you placed `zest.releaser`. Or, if you installed `zest.releaser` globally, your egg-with-entrypoint has to be globally installed, too.

7.2 Comments about data dict items

Your entry point gets a data dictionary: the items you get in that dictionary are documented below. Some comments about them:

- Not all items are available. If no history/changelog file is found, there won't be any `data['history_lines']` either.
- Items that are templates are normal python string templates. They use dictionary replacement: they're actually passed the same data dict. For instance, `prerelease's data['commit_message']` is by default `Preparing release %(new_version)s`. A "middle" entry point could modify this template to get a different commit message.

7.3 Prerelease data dict items

commit_msg Message template used when committing

history_file Filename of history/changelog file (when found)

history_header Header template used for 1st history header

history_lines List with all history file lines (when found)

name Name of the project being released

new_version New version (so 1.0 instead of 1.0dev)

original_version Version before prereleasing (e.g. 1.0dev)

today Date string used in history header

workingdir Original working directory

7.4 Release data dict items

name Name of the project being released

tag_already_exists Internal detail, don't touch this :-)

tagdir Directory where the tag checkout is placed (*if* a tag checkout has been made)

version Version we're releasing

workingdir Original working directory

7.5 Postrelease data dict items

commit_msg Message template used when committing

dev_version New development version with dev marker (so 1.1.dev0)

dev_version_template Template for dev version number

history_header Header template used for 1st history header

name Name of the project being released

new_version New development version (so 1.1)

nothing_changed_yet First line in new changelog section

workingdir Original working directory

CHANGELOG FOR ZEST.RELEASER

8.1 3.43 (2013-02-04)

- Added `--no-input` commandline option for running automatically without asking for input. Useful when started from some build tool. See the documentation at the end of <http://zestreleaser.readthedocs.org/en/latest/uploading.html> . [reinout, based upon a patch by j-san]

8.2 3.42 (2013-01-07)

- When finding multiple version, changes or history files, pick the one with the shortest path. [maurits]
- Support project-specific hooks listed in `setup.cfg`. [iguananaut]

8.3 3.41 (2012-11-02)

- Getting the version from `setup.py` can give a traceback if the `setup.py` has an error. During prerelease this would result in a proposed version of 'Traceback'. Now we print the traceback and quit. [maurits]

8.4 3.40 (2012-10-13)

- Support `svn` (1.7+) checkouts that are not directly in the root. Only applies when someone checks out a whole tree and wants to release one of the items in a subdirectory. Fixes #27.

8.5 3.39 (2012-09-26)

- Only search for files in version control. This is when finding a history file or `version.txt` file. We should not edit files that are not in our package. Fixes issue #22. [maurits]

8.6 3.38 (2012-09-25)

- Fixed svn tag extraction on windows: a `\r` could end up at the end of every tag name. Thanks Wouter Vanden Hove for reporting it!
- Small fixes to the developers documentation and to the automatic [travis CI](#) tests configuration.

8.7 3.37 (2012-07-14)

- Documentation update! Started sphinx documentation at [zestreleaser.readthedocs.org](#). Removed documentation from the README and put it into sphinx.
- Actually ask if the user wants to continue with the release when there is no MANIFEST.in. We asked for a yes/no answer, but the question was missing. [maurits]

8.8 3.36 (2012-06-26)

- Improved changes/history file detection and fixed the documentation at this point. We now recognize CHANGES, HISTORY and CHANGELOG with `.rst`, `.txt`, `.markdown` and with no extension.
- Set up [travis CI](#) integration. Our tests pass on python 2.5, 2.6 and 2.7.

8.9 3.35 (2012-06-21)

- When checking for recommended files, ask if the user wants to continue when we suspect the created PyPI release may be broken. See issue #10. [maurits]
- Preserve existing EOL in `setup.py` and `history` file (See <http://docs.python.org/tutorial/inputoutput.html#reading-and-writing-files>) [tom_gross]

8.10 3.34 (2012-03-20)

- In the warning about a missing MANIFEST.in file, also suggest to install `setup-tools_subversion/git`, etc. Fixes issue #4. [maurits]

8.11 3.33 (2012-03-20)

- Fix python 2.4 issues with tarfile by always creating a zip file. Formerly we would only do this when using python2.4 for doing the release, but a tarball sdist created by python2.6 could still break when the end user is using python 2.4. [kiorky]

8.12 3.32 (2012-03-09)

- In prerelease recommend the user to add a MANIFEST.in file. See <http://docs.python.org/distutils/sourcedist.html> for more info. [maurits]

8.13 3.31 (2012-02-23)

- Fixed test for unadvised egg_info commands on tag, which could result in a ConfigParser error. [maurits]

8.14 3.30 (2011-12-27)

- Added some more PyPI classifiers. Tested with Python 2.4, 2.4, 2.6, and 2.7. [maurits]
- Moved changes of 3.15 and older to docs/HISTORY.txt. [maurits]
- Added GPL license text in the package. [maurits]
- Updated README.txt. Added MANIFEST.in. [maurits]

8.15 3.29 (2011-12-27)

- In postrelease create a version number like 1.0.dev0. See <http://www.python.org/dev/peps/pep-0386> [maurits]
- Offer to cleanup setup.cfg on the tag when releasing. You do not want tag_build or tag_svn_revision options in a release usually. [maurits]
- For convenience also print the tag checkout location when only doing a release (instead of a fullrelease). [maurits]

8.16 3.28 (2011-11-18)

- Git: in pre/postrelease only check for uncommitted changes in files that are already tracked. [maurits]

8.17 3.27 (2011-11-12)

- Postrelease now offers (=asks) to push your changes to the server if you're using hg or git.
- Support for some legacy projects, often converted from CVS, have multiple subprojects under a single trunk. The trunk part from the top level project isn't erroneously stripped out anymore. Thanks to Marc Sibson for the fix.

8.18 3.26 (2011-11-01)

- Added sanity check before doing a prerelease so you are warned when you are about to commit on a tag instead of a branch (or trunk or master). [maurits]

8.19 3.25 (2011-10-28)

- Removed special handling of subversion lower than 1.7 when searching for the history/changes file. In corner cases it may be that we find a wrong HISTORY.txt or CHANGES.txt file when you have it buried deep in your directory structure. Please move it to the root then, which is the proper place for it. [maurits]
- Fixed finding a history/changes file that is in a sub directory when using subversion 1.7 or higher or bazaar. [maurits]

8.20 3.24 (2011-10-19)

- Note: you may need to install `setuptools_subversion` when you use subversion 1.7. If you suddenly start missing files in the sdist you upload to PyPI you definitely need it. Alternatively: set up a proper MANIFEST.in as that method works with any version control system. [maurits]
- Made compatible with subversion 1.7 (the only relevant change is in the code that checks if a tags or tag directory already exists). Earlier versions of subversion are of course still supported. [maurits]
- Code repository moved to github: <https://github.com/zestsoftware/zest.releaser> [maurits]

8.21 3.23 (2011-09-28)

- Fixed opening the html long description in `longtest` on Mac OS X Lion or python2.7 by using a `file://` url. Fixes <https://bugs.launchpad.net/zest.releaser/+bug/858011> [maurits]

8.22 3.22 (2011-05-05)

- Allow specifying a tag on the command line when using `lasttaglog` or `lasttagdiff`, to show the log or diff since that tag instead of the latest. Useful when you are on a branch and the last tag was from trunk. [maurits]

8.23 3.21 (2011-04-20)

- Added `lasttaglog` command that list the log since the last tag. [maurits]
- Fix Mercurial (hg) support. As `spreaded_internal` should be set to False (as it happens with git) [erico_andrei]
- Accept a twiddle (or whatever '~' is called) when searching for headers in a changelog; seen in some packages (at least `zopeskel.dexterity`). [maurits]

8.24 3.20 (2011-01-25)

- Also allowing CHANGES.rst and CHANGES.markdown in addition to CHANGES.txt.

8.25 3.19 (2011-01-24)

- No longer refuse to register and upload a package on pypi if it is not there yet, forcing people to do this manually the first time. Instead, we ask the question and simply have ‘No’ as the default answer. If you specify an answer, we require exactly typing ‘yes’ or ‘no’. The idea is still to avoid making it too easy to release an internal package on pypi by accident. [maurits]

8.26 3.18 (2010-12-08)

- Added `--non-interactive--` option to the `svn diff` command used in `lasttagdiff`. This makes it usable in cronjobs and post-commit hooks. Fixes <https://bugs.launchpad.net/zest.releaser/+bug/687530>

8.27 3.17 (2010-11-17)

- When the package that is being released neither has a `setup.py` nor a `setup.cfg`, use No as default answer for creating a checkout of the tag. [maurits]

8.28 3.16 (2010-11-15)

- For (pypi) output, also show the first few lines instead of only the last few. [maurits]
- See if `pyirc` or `setup.cfg` has a `[zest.releaser]` section with option `release = yes/no`. During the release stage, this influences the default answer when asked if you want to make a checkout of the tag. The default when not set, is ‘yes’. You may want to set this to ‘no’ if most of the time you only make releaser of internal packages for customers and only need the tag. [maurits]
- Specify bazaar (bzt) tag numbers using the ‘tag’ revision specifier (like ‘tag:0.1’) instead of only the tag number (0.1) to add compatibility with earlier bzt versions (tested with 2.0.2). [maurits]

8.29 3.15 (2010-09-10)

- Read pypi config not only from the `.pyirc` file, but also from the `setup.cfg` file of the package. Patch by Erico Andrei. [maurits]

8.30 3.14 (2010-08-26)

- experimental support for git svn tagging, fully test-covered [chaoflow]
- fail if no tag was created, not test-covered [chaoflow]
- `svn available_tags` method: intercept ‘Repository moved’ note in `svn info` and stop processing then. [maurits]

8.31 3.13 (2010-08-16)

- Fixed check that tested whether a package was already available on pypi, as the pypi implementation changed slightly. We now just check for a 404 status. Patch by Wolfgang Schnerring. [maurits]

8.32 3.12 (2010-07-22)

- Added extra entry point for the release step: `after_checkout`. When this is run, the middle entry point has been handled, the tag has been made, a checkout of that tag has been made and we are now in that checkout directory. Idea: Jan-Wijbrand Kolman. [maurits]
- Fix: in the `zest.releaser.releaser.after` entry point data, pass the `'tagdir'` value (if a checkout has been made). Patch by Wolfgang Schnerring, thanks! [maurits]
- Fixed tests to also pass with slightly newer git. [maurits]

8.33 3.11 (2010-06-25)

- Small tweak: allowing `zc.rst2`'s `"rst2 html"` in addition to `docutils`' own `"rst2html"`.

8.34 3.10 (2010-06-15)

- Fix : when running `'release'` with python2.6 against a private egg server, the `distutils 'register'` command would run against PyPI while `'upload'` command would run against private server. (`-r` option needs to be stated twice) [gotcha]

8.35 3.9 (2010-06-11)

- Again at the end of a fullrelease report the location of the directory containing the checkout of the tag, if it has been made. [maurits]

8.36 3.8 (2010-05-28)

- Also allowing `CHANGES` in addition to `HISTORY.txt` and `CHANGES.txt` as a history filename. Keeps several Django packages happy.

8.37 3.7 (2010-05-07)

- Added support for bazaar. Fixes <https://bugs.launchpad.net/zest.releaser/+bug/490816> [menesis]

8.38 3.6 (2010-04-13)

- A `version='1.0'`, string inside the `setup()` call no longer has non-pep8 spaces around the `=`. Fixes <https://bugs.launchpad.net/zest.releaser/+bug/562122> [reinout]
- Got rid of ugly `setup.py` hack with `UltraMagicString` that was meant to avoid encoding errors when registering this package at pypi but which was not working for python2.4 (at least with `collective.dist`). Only `ascii` is allowed in the `long_description` if you want to avoid problems at one point or another. [maurits]

8.39 3.5 (2010-02-26)

- Treat `CHANGES.txt` and `HISTORY.txt` the same: the first that is found in a directory is chosen for changing, instead of first looking everywhere for a `HISTORY.txt` and then for a `CHANGES.txt`. [maurits]

8.40 3.4 (2010-02-02)

- Always build zip files if using python2.4 [do3cc]
- bugfix: added `'spreaded_internal'` property to `BaseVersionControl` objects, so `filefind()` does not exclude a directory just because there is no `'.git'` folder in it. It still excludes directory where there is no `'.svn'` folder in SVN repositories. [vincent]

8.41 3.3 (2009-12-29)

- Fixed test failures when run on a computer with a new style pypi config. We now always use an old style config when running the tests. [maurits]
- Fixed the release command for hg 1.1 (e.g. Ubuntu 9.04). [maurits]

8.42 3.2 (2009-12-22)

- Replaced `commands.getoutput()` with a `system()` function grabbed from `buildout` on suggestion by Adam Groszer. Goal: make `zest.releaser` work also on windows.
- Improved entry point documentation.
- Added launchpad bugtracker at <https://bugs.launchpad.net/zest.releaser> (and pointing at that in the documentation).

8.43 3.1 (2009-11-27)

- Added documentation for entry points. [reinout]

8.44 3.0 (2009-11-27)

- Added support for extension by means of entry points. There is no documentation that advertises it yet as I want to treat it as experimental till I've used it a few times. [reinout]

8.45 2.12 (2009-11-26)

- Fixed mercurial sdist creation. [reinout]
- A missing history file does not result anymore in a keyerror in prerelease. [reinout]
- Added lots of test output normalization so that errors aren't hidden by the large number of . . . in the doctests. [reinout]

8.46 2.11 (2009-11-25)

- Added /tag besides /tags for subversion [gotcha]
- Fixed tests failures. [gotcha]

8.47 2.10 (2009-10-22)

- Added support for git. [reinout]
- Lots of internal refactoring and small fixes. [reinout]
- Started tests. zest.releaser went from 0 to 94% coverage. [reinout]

8.48 2.9.3 (2009-09-22)

- Uploading to multiple package indexes should now work in python2.6 (though ironically it now does not work for me on python2.4, but that has nothing to do with zest.releaser.) Added documentation for this. [maurits]
- Make sure the next version suggestion for 1.0rc6 is 1.0rc7. [maurits]
- In subversion, first try to get the package from the setup.py before falling back to the svn info, just like for mercurial. This fixes the problem that e.g. Products.feedfeeder was not recognized as being on pypi as the svn directory name was feedfeeder. [maurits]

8.49 2.9.2 (2009-09-17)

- Umlauts in a changelog don't break the logger anymore when using python2.6.2 when the umlauts turn up in the diff. This is due to a 2.6.2 regression bug, see <http://bugs.python.org/issue5170>. Should be fixed in 2.6.3 when it comes out. [reinout]
- (Release 2.9 and 2.9.1 are unreleased because of a setuptools bug with, sigh, non-ascii characters which made a dirty setup.py hack necessary). [reinout]

8.50 2.8 (2009-08-27)

- Fixed the release command when used in a french environment. In French “svn info” returns ‘URL :’, not ‘URL:’. [vincentfretin]

8.51 2.7 (2009-07-08)

- Before asking setup.py for its version or name, first run egg_info, as that may get rid of some warnings that otherwise end up in the extracted version or name, like UserWarnings. [maurits]

8.52 2.6 (2009-05-25)

- Small change: the questions don’t print a newline anymore after the question (and before the user pressed enter). This makes it clearer if enter has been pressed yet. Suggestion by jkolman. [reinout]

8.53 2.5 (2009-05-20)

- Revert to previous behaviour: when a package has not been released yet on pypi, decline to register it: the first time should be deliberate and manual, to avoid accidentally uploading client packages to pypi. [maurits]

8.54 2.4 (2009-05-15)

- Factored release.py out into a new pypi.py, solving a few possible problems with missing or mis-configured .pypirc files. [maurits]

8.55 2.3 (2009-05-11)

- Fixed release script when the .pypirc file does not contain a distutils section or that section does not contain a index-servers option. [maurits]

8.56 2.2 (2009-05-11)

- postrelease: suggestion for next version after 1.1.19 is not 1.1.110, but 1.1.20. [maurits]
- Make it work with collective.dist (mregister/mupload) [WouterVH] see <http://plone.org/documentation/tutorial/how-to-upload-your-package-to-plone.org/installing-collective.dist>

8.57 2.1 (2009-04-09)

- Fix `lasttagdiff` command to work with Mercurial by truncating the ‘+’ character from the revision id, since that only indicates uncommitted changes exist.
- Make sure we find `package/name/HISTORY.txt` before we find `docs/HISTORY.txt`. [maurits]
- Fixed checking for `self.internal_filename`: we would incorrectly check (‘.’, ‘s’, ‘v’, ‘n’) instead of ‘.svn’. [maurits]

8.58 2.0 (2009-04-01)

- Added `tag_url` method to get `lasttagdiff` (and `zest.stabilizer`) working again. [maurits]
- Merged `kteague-multi-vcs` branch with, woohoo, mercurial support! [reinout]
- Mercurial support by Kevin Teague. [kteague]
- `postrelease` put a space in the new version number in `setup.py` (between version number and dev). Removed this space as it is not necessary (in best case). [icemac]

8.59 1.13 (2009-03-17)

- Also looking for `CHANGES` in addition to `HISTORY.txt` and `CHANGES.txt` as some packages use that convention. [reinout]
- Added `lasttagdiff` command that shows the diff between the last release and the currently committed trunk. Handy for checking whether the changelog is up to date. [reinout]

8.60 1.12 (2009-03-17)

- When doing a `fullrelease` and if the release step made a checkout of the tag into an temp directory, that temp directory is again printed after `fullrelease` finishes. Otherwise you’ve got to do a lot of scrolling. [reinout]

8.61 1.11 (2009-03-04)

- When the found history file contains no version headings, look for a second history file: more than once I have the standard `docs/HISTORY.txt` that paster creates and I just add a pointer there to the real `package/name/HISTORY.txt`. [maurits]

8.62 1.10 (2009-02-25)

- A “`version = ‘1.0’`,” in `setup.py` is now also rewritten correctly. Previously just a `version = ‘1.0’` would be injected, so without indentation and comma. [reinout]
- Ask before checking out the tag. Sometimes the checkout is huge and you know you don’t want it. You don’t get asked for a pypi upload, though if you don’t check out the tag. [reinout]

8.63 1.9 (2009-02-24)

- ‘release’ now also makes a tag checkout in a temporary directory. [Reinout]
- Made ‘longtest’ work on Linux as there the command is ‘rst2html’ and apparently on the Mac it is ‘rst2html.py’. [maurits]

8.64 1.8 (2009-02-23)

- Added ‘longtest’ command that renders a setup.py’s long description and opens it in a web browser. [reinout]

8.65 1.7 (2009-02-16)

- Supporting alternative history version header format: ‘version - date’. [reinout]

8.66 1.6 (2009-02-14)

- Patch by Michael Howitz: sys.executable is used instead of a string that doesn’t work on every system. [reinout]

8.67 1.5 (2009-02-11)

- Changed y/n into Y/n, so defaulting to ‘yes’. [reinout]
- Improved the documentation. [reinout]
- When a yes/no question is asked, do not treat ‘no’ as the default but explicitly ask for an input – it was too easy to press enter and wrongly expect ‘yes’ as default. [maurits]

8.68 1.4 (2008-10-23)

- Fixed missing import of utils. [maurits]

8.69 1.3 (2008-10-23)

- Moved stabilize script to zest.stabilizer so that zest.releaser is just for releasing individual packages. Nice, tidy, reusable. [reinout]
- Allowing ‘-v’ option on all commands: it gives you debug-level logging. [reinout]

8.70 1.2 (2008-10-16)

- We now prefer the version from setup.py over any version.txt file found. When getting or changing the version we get/change the setup.py version when it differs from the found version.txt version. [maurits]

8.71 1.1 (2008-10-15)

- Cleaned out zest-specific stuff. Cleaned up ‘release’. [reinout]

8.72 1.0 (2008-10-15)

- Stabilize looks up the most recent tag of our development packages and uses gp.svndevelop to allow svn checkouts as development eggs. [reinout]
- Do not look for version.txt in directories that are not handled by subversion. Use case: Products.feedfeeder, which has a buildout.cfg and so can have a parts directory with lots of version.txt files... [maurits]

8.73 0.9 (2008-10-02)

- release: offer to register and upload the egg to the cheese shop. After that you still have the option to upload to our own tgz server. [maurits]
- postrelease: for the suggestion of a new version simply try add 1 to the last character in the version; the previous logic failed for example for ‘1.0b3’. [maurits]
- prerelease: ask user to enter next version (give him a suggestion). Handy when you want to change ‘1.0b3 dev’ into ‘1.0’. [maurits]
- Started ‘stabilize’. [reinout]

8.74 0.8 (2008-09-26)

- fullrelease: change back to the original directory after each pre/actual/post release step. [maurits]
- release: switch back to original directory when ready to fix ‘commit to tag’ error. [maurits]
- prerelease: quit when no version is found. [maurits]
- Reverted sleeping fix from 0.7 as it did not work. [maurits]

8.75 0.7 (2008-09-26)

- fullrelease: hopefully fix a ‘commit on tag’ bug by sleeping three seconds before doing the post release. [maurits]

8.76 0.6 (2008-09-26)

- Added fullrelease script that does a prerelease, actual release and post release in one go. [maurits]

8.77 0.5 (2008-09-26)

- Factored part of prerelease.check_version() out into utils.cleanup_version(). We now use that while setting the version in the history during postrelease. [maurits]
- Add newline at the end of the generated version.txt. [maurits]

8.78 0.4 (2008-09-26)

- Made the logging userfriendly.

8.79 0.3 (2008-09-26)

- Postrelease: Better injection of history. Various other robustness fixes.

8.80 0.2 (2008-09-26)

- postrelease: added suggestion for new version (a plain enter is enough to accept it). [reinout]
- prerelease: ask before changing version + solidified regex for heading detection. [reinout]
- prerelease: detect non-development versions better and change them. [maurits]
- prerelease: made the commit message read: 'Preparing release xxx'. [maurits]
- postrelease: made the new version something like '1.0 dev'. [maurits]
- postrelease: we now add some lines to the history now. [maurits]
- prerelease: try changing the version to a non-development version, stripping off something like '(...)'. [maurits]
- release: Refactored so release.py has the 'main' function required by setup.py. [maurits]

8.81 0.1 (2008-09-24)

- Got a basic version of the prerelease script working (version check, history file updating, committing). [reinout]
- Started by copying the guidelines script. [reinout]