
gilt Documentation

Release unknown

AUTHORS.md

Mar 17, 2020

CONTENTS

1 Quick Start	3
2 Documentation	5
3 Alternatives	7
4 License	9
5 Usage:	11
6 Molecule	13
7 Similar Tools	15
7.1 Testing	15
7.2 Contributing	15
7.2.1 Testing	15
7.2.2 Formatting	16
7.3 Development	16
7.3.1 Branches	16
7.3.2 Release Engineering	16
7.3.3 Roadmap	17
7.4 History	17
7.4.1 1.2.3	17
7.4.2 1.2.2	17
7.4.3 1.2.1	18
7.4.4 1.2	18
7.4.5 1.1	18
7.4.6 1.0	18
7.5 Credits	18
7.5.1 Development Leads	18
7.5.2 Core Committers	18
7.5.3 Contributors	18
7.6 Autodoc	18
7.6.1 Config	18
7.6.2 Git	19
7.6.3 Util	19
8 Indices and tables	21
Python Module Index	23

gilt - A GIT layering tool.

QUICK START

Install `gilt` using `pip`:

```
$ pip install python-gilt
```

Overlay remote repositories and/or files into the destinations provided.

```
$ gilt --debug overlay
```


DOCUMENTATION

<https://gilt.readthedocs.io/>

ALTERNATIVES

- [Repo](#)
- [Git submodules](#)
- [Git subtree](#)
- [Go Gilt](#) - An indirect port of Gilt to Golang (WIP)

LICENSE

MIT

The logo is licensed under the [Creative Commons NoDerivatives 4.0 License](#). If you have some other use in mind, contact us.

USAGE:

Overlay a remote repository into the destination provided.

Listing 1: gilt.yml

```
- git: https://github.com/retr0h/ansible-etcd.git
  version: master
  dst: roles/retr0h.ansible-etcd/
```

```
$ gilt overlay
```

Overlay files from a remote repository into the destinations provided.

Listing 2: gilt.yml

```
- git: https://github.com/lorin/openstack-ansible-modules.git
  version: master
  files:
    - src: "*_manage"
      dst: library/
    - src: nova_quota
      dst: library/
    - src: neutron_router
      dst: library/neutron_router.py
```

```
$ gilt overlay
```

Overlay a directory from a remote repository into the destination provided.

Listing 3: gilt.yml

```
- git: https://github.com/blueboxgroup/ursula.git
  version: master
  files:
    - src: roles/logging
      dst: roles/blueboxgroup.logging/
```

```
$ gilt overlay
```

Optionally, override gilt's cache location (defaults to ~/.gilt):

```
$ export GILT_CACHE_DIRECTORY=~/.my-gilt-cache
```

Overlay files and a directory and run post-overlay commands.

Listing 4: gilt.yml

```
- git: https://github.com/example/subproject.git
version: master
files:
  - src: subtool/test
    dst: ext/subtool.test/
    post_commands:
      - make

- git: https://github.com/example/subtool2.git
version: master
dst: ext/subtool2/
post_commands:
  - make
```

```
$ gilt overlay
```

Display the git commands being executed.

```
$ gilt --debug overlay
```

Use an alternate config file (default *gilt.yml*).

```
$ gilt --config /path/to/gilt.yml overlay
```


MOLECULE

Integrates with [Molecule](#) as an [Ansible Galaxy](#) CLI replacement. Update *molecule.yml* with the following.

```
---  
dependency:  
  name: shell  
  command: gilt overlay
```


SIMILAR TOOLS

- [Ansible Galaxy](#)

7.1 Testing

```
$ pip install tox
$ tox
```

7.2 Contributing

- We are interested in various different kinds of improvement for Gilt; please feel free to raise an [Issue](#) if you would like to work on something major to ensure efficient collaboration and avoid duplicate effort.
- Create a topic branch from where you want to base your work.
- Check for unnecessary whitespace with `git diff --check` before committing.
- Make sure you have added tests for your changes.
- Run all the tests to ensure nothing else was accidentally broken.
- Reformat the code by following the formatting section below.
- Submit a pull request.

7.2.1 Testing

Dependencies

Install the test framework [Tox](#).

```
$ pip install tox
```

Unit

Unit tests are invoked by `Tox`.

```
$ tox
```

7.2.2 Formatting

The formatting is done using `Black`.

From the root for the project, run:

```
$ tox -e format
```

7.3 Development

- Please read the *Contributing* guidelines.

7.3.1 Branches

- The `master` branch is stable. Major changes should be performed elsewhere.

7.3.2 Release Engineering

Pre-release

- Edit the *History*.
- Follow the *Testing* steps.

Release

Gilt follows `Semantic Versioning`.

Tag the release and push to github.com

```
$ git tag v2.0.0  
$ git push --tags
```

Upload to PyPI

- Upload to [PyPI](#).

```
$ tox -e build-dists
$ tox -e publish-dists
```

Post-release

- Comment/close any relevant [Issues](#).

7.3.3 Roadmap

- See [Issues](#) on Github.com.

7.4 History

7.4.1 1.2.3

- Add bitbucket support.
- Add CI.
- Correct hard coded paths in tests.
- Remove black skip-string-normalization.
- Add coverage.
- Switched to setuptools vs pbr.
- Switched to black off yapf.
- Added hacking for extra lint.
- Corrected docstrings.

7.4.2 1.2.2

- Cleanup dst prior to checkout-index.
- Fix ambiguous variable usage.
- Fixed deprecation of getfuncargvalue.
- Remove use of deprecated pytest.config.
- Fixed test failure with GILT_CACHE_DIRECTORY.
- Fix tox failure due to use of relative path with -cov.

7.4.3 1.2.1

- Use proper package name for pbr.

7.4.4 1.2

- Add option to override gilt's default cache dir.

7.4.5 1.1

- Add support for running commands after a sync has happened.
- Added python 3 support.
- Suppress doc building warnings.
- Only fetch when branch/tag/commit not on local repo.
- Determine branch in a smarter way.

7.4.6 1.0

- Initial release.

7.5 Credits

7.5.1 Development Leads

- John Dewey (@retr0h)

7.5.2 Core Committers

7.5.3 Contributors

```
$ git shortlog -s | cut -c8-
```

7.6 Autodoc

7.6.1 Config

exception `gilt.config.ParseError`

Error raised when a config can't be loaded properly.

`gilt.config.config` (*filename*)

Construct *Config* object and return a list.

Parse filename A string containing the path to YAML file.

Returns list

7.6.2 Git

`gilt.git.clone` (*name, repository, destination, debug=False*)

Clone the specified repository into a temporary directory and return None.

Parameters

- **name** – A string containing the name of the repository being cloned.
- **repository** – A string containing the repository to clone.
- **destination** – A string containing the directory to clone the repository into.
- **debug** – An optional bool to toggle debug output.

Returns None

`gilt.git.extract` (*repository, destination, version, debug=False*)

Extract the specified repository/version into the directory and return None.

Parameters

- **repository** – A string containing the path to the repository to be extracted.
- **destination** – A string containing the directory to clone the repository into. Relative to the directory `gilt` is running in. Must end with a `'/'`.
- **version** – A string containing the branch/tag/sha to be exported.
- **debug** – An optional bool to toggle debug output.

Returns None

`gilt.git.overlay` (*repository, files, version, debug=False*)

Overlay files from repository/version into the directory and return None.

Parameters

- **repository** – A string containing the path to the repository to be extracted.
- **files** – A list of `FileConfig` objects.
- **version** – A string containing the branch/tag/sha to be exported.
- **debug** – An optional bool to toggle debug output.

Returns None

7.6.3 Util

`gilt.util.build_sh_cmd` (*cmd, cwd=None*)

Build a `sh.Command` from a string.

Parameters

- **cmd** – String with the command to convert.
- **cwd** – Optional path to use as working directory.

Returns `sh.Command`

`gilt.util.copy` (*src, dst*)

Handle the copying of a file or directory.

The destination basedir `_must_` exist.

Parameters

- **src** – A string containing the path of the source to copy. If the source ends with a `'/'`, will become a recursive directory copy of source.
- **dst** – A string containing the path to the destination. If the destination ends with a `'/'`, will copy into the target directory.

Returns None

`gilt.util.print_info(msg)`

Print the given message to STDOUT.

`gilt.util.print_warn(msg)`

Print the given message to STDOUT in YELLOW.

`gilt.util.run_command(cmd, debug=False)`

Execute the given command and return None.

Parameters

- **cmd** – A *sh.Command* object to execute.
- **debug** – An optional bool to toggle debug output.

Returns None

`gilt.util.saved_cwd()`

Context manager to restore previous working directory.

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

g

`gilt.config`, 18
`gilt.git`, 19
`gilt.util`, 19

INDEX

B

`build_sh_cmd()` (in module *gilt.util*), 19

C

`clone()` (in module *gilt.git*), 19

`config()` (in module *gilt.config*), 18

`copy()` (in module *gilt.util*), 19

E

`extract()` (in module *gilt.git*), 19

G

gilt.config (module), 18

gilt.git (module), 19

gilt.util (module), 19

O

`overlay()` (in module *gilt.git*), 19

P

`ParseError`, 18

`print_info()` (in module *gilt.util*), 20

`print_warn()` (in module *gilt.util*), 20

R

`run_command()` (in module *gilt.util*), 20

S

`saved_cwd()` (in module *gilt.util*), 20