
gilt Documentation

Release None

AUTHORS.md

November 29, 2016

1	Quick Start	3
2	Documentation	5
3	License	7
4	Usage:	9
5	Molecule	11
6	Testing	13
7	Similar Tools	15
7.1	Contributing	15
7.1.1	Testing	15
7.1.2	Formatting	16
7.2	History	16
7.2.1	1.0	16
7.3	Credits	16
7.3.1	Development Leads	16
7.3.2	Core Committers	16
7.3.3	Contributors	16
7.4	Autodoc	16
7.4.1	Config	16
7.4.2	Git	16
7.4.3	Util	17
8	Indices and tables	19
	Python Module Index	21

gilt - A GIT layering tool.

Quick Start

Install gilt using pip:

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

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

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


Documentation

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

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 4.1: 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 4.2: 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 4.3: gilt.yml

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

```
$ 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: girt overlay
```


Testing

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

Similar Tools

- Reponymous
- Ansible Galaxy

7.1 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.

7.1.1 Testing

Dependencies

Install the test framework [Tox](#).

```
$ pip install tox
```

Unit

Unit tests are invoked by [Tox](#).

```
$ tox
```

7.1.2 Formatting

The formatting is done using [YAPF](#).

From the root for the project, run:

```
$ tox -e syntax
```

7.2 History

7.2.1 1.0

- Initial release.

7.3 Credits

7.3.1 Development Leads

- John Dewey (@retr0h)

7.3.2 Core Committers

7.3.3 Contributors

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

7.4 Autodoc

7.4.1 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

7.4.2 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 given 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 the specified repository/version into the given 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

7.4.3 Util

`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(*args, **kwds)`

Context manager to restore previous working directory.

Indices and tables

- genindex
- modindex
- search

g

`gilt.config`, 16
`gilt.git`, 16
`gilt.util`, 17

C

clone() (in module `gilt.git`), 16
config() (in module `gilt.config`), 16
copy() (in module `gilt.util`), 17

E

extract() (in module `gilt.git`), 17

G

`gilt.config` (module), 16
`gilt.git` (module), 16
`gilt.util` (module), 17

O

overlay() (in module `gilt.git`), 17

P

`ParseError`, 16
print_info() (in module `gilt.util`), 17
print_warn() (in module `gilt.util`), 17

R

run_command() (in module `gilt.util`), 17

S

saved_cwd() (in module `gilt.util`), 18