
fabric_package_management Documentation

Release

Author

Jul 02, 2017

Contents

1	Code Documentaion:	3
1.1	fabric_package_management package	3
1.1.1	apt module	3
2	Example	7
3	Indices and tables	9
	Python Module Index	11

As the name implies, fabric-package-management is a collection of [Fabric](#) tasks for package management.

fabric_package_management package

apt module

`fabric_package_management.apt.autoclean` (*use_sudo=True, verbose=True*)

Like *clean*, but only removes package files that can no longer be downloaded.

Parameters

- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

`fabric_package_management.apt.autoremove` (*assume_yes=True, use_sudo=True, verbose=True*)

Parameters

- **assume_yes** (*bool*) – If *True*, Apt will assume “yes” as answer to all prompts and run non-interactively. (Default: *True*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

`fabric_package_management.apt.build_dep` (*package, assume_yes=True, use_sudo=True, verbose=True*)

Install the build dependencies for a given source package.

Parameters

- **package** (*str*) – The package whose build dependencies will be installed.
- **assume_yes** (*bool*) – If *True*, Apt will assume “yes” as answer to all prompts and run non-interactively. (Default: *True*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)

- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

`fabric_package_management.apr.check_version_available (package, version)`

`fabric_package_management.apr.clean (use_sudo=True, verbose=True)`

Clears out retrieved package files.

Parameters

- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

`fabric_package_management.apr.dist_upgrade (assume_yes=True, use_sudo=True, verbose=True)`

Same as *upgrade*, but Apt will attempt to intelligently handle changing dependencies, installing new dependencies as needed.

Parameters

- **assume_yes** (*bool*) – If *True*, Apt will assume “yes” as answer to all prompts and run non-interactively. (Default: *True*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

`fabric_package_management.apr.install (packages, assume_yes=True, no_install_recommends=False, install_suggests=False, use_sudo=True, verbose=True, force_yes=False)`

Install packages on the remote host via Apt.

Parameters

- **packages** (*list or str*) – The packages to install.
- **no_install_recommends** (*bool*) – Apt will not consider recommended packages as a dependencies for installing. (Default: *True*)
- **install_suggests** (*bool*) – Apt will consider suggested packages as a dependency for installing. (Default: *False*)
- **assume_yes** (*bool*) – If *True*, Apt will assume “yes” as answer to all prompts and run non-interactively. (Default: *True*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)
- **force_yes** (*bool*) – add the *-force-yes* apt-get option. (Default: *False*)

`fabric_package_management.apr.installed (package, use_sudo=True)`

Check if a package is installed on the system.

Returns *True* if installed, *False* if it is not.

Parameters

- **package** (*str*) – The package to check if installed.
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *False*)

`fabric_package_management.apr.reboot_required (use_sudo=False, verbose=False)`

Check if a reboot is required after installing updates.

Returns *True* if a reboot is required, *False* if not.

Parameters

- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *False*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *False*)

```
fabric_package_management.apt.remove(packages, purge=False, assume_yes=True,  
                                     use_sudo=True, verbose=True)
```

Remove a package or list of packages from the remote host.

Parameters

- **packages** (*list or str*) – The packages to install.
- **purge** (*bool*) – If *True* any configuration files are deleted too. (Default: *False*)
- **assume_yes** (*bool*) – If *True*, Apt will assume “yes” as answer to all prompts and run non-interactively. (Default: *True*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

```
fabric_package_management.apt.source(package, download_only=False, use_sudo=False, ver-  
                                     bose=True)
```

Download a given source package.

Parameters

- **package** (*str*) – The source package to download.
- **download_only** (*bool*) – If *True*, the source package will not be unpacked. (Default: *False*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *False*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

```
fabric_package_management.apt.update(use_sudo=True, verbose=True, source_name=None)
```

Update Apt’s package index files on the remote host.

Parameters

- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)
- **source_name** (*str*) – If set, update only the sources defined in that *sources.list.d* file.

```
fabric_package_management.apt.upgrade(assume_yes=True, use_sudo=True, verbose=True)
```

Install the newest versions of all packages on the remote host.

Parameters

- **assume_yes** (*bool*) – If *True*, Apt will assume “yes” as answer to all prompts and run non-interactively. (Default: *True*)
- **use_sudo** (*bool*) – If *True*, will use *sudo* instead of *run*. (Default: *True*)
- **verbose** (*bool*) – If *False*, hide all output. (Default: *True*)

CHAPTER 2

Example

```
#!/usr/bin/python

from fabric.api import task, prompt, env
from fabric.context_managers import cd
from fabric.operations import reboot

from fabric_package_management import apt

@task()
def run():
    apt.update()
    apt.upgrade()
    apt.install(['bpython', 'git'])
    with cd('/tmp'):
        apt.source("python-libcloud", download_only=True)
    apt.remove('bpython', purge=True)
    apt.autoremove()
    apt.autoclean()
    if apt.reboot_required():
        prompt("Reboot required. Initiate now?\nYes/No?",
              "response",
              default="No",
              validate=r'yes|Yes|YES|no|No|NO')
        if env.response.lower() == "yes":
            reboot()
```


CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

f

`fabric_package_management`, 5

`fabric_package_management.apt`, 3

A

autoclean() (in module fabric_package_management.apt),
3

autoremove() (in module fabric_package_management.apt), 3

B

build_dep() (in module fabric_package_management.apt), 3

C

check_version_available() (in module fabric_package_management.apt), 4

clean() (in module fabric_package_management.apt), 4

D

dist_upgrade() (in module fabric_package_management.apt), 4

F

fabric_package_management (module), 5

fabric_package_management.apt (module), 3

I

install() (in module fabric_package_management.apt), 4

installed() (in module fabric_package_management.apt),
4

R

reboot_required() (in module fabric_package_management.apt), 4

remove() (in module fabric_package_management.apt), 5

S

source() (in module fabric_package_management.apt), 5

U

update() (in module fabric_package_management.apt), 5

upgrade() (in module fabric_package_management.apt),
5