

---

# **fabric\_package\_management Documentation**

*Release*

**Author**

**Jun 30, 2017**



---

## Contents

---

|   |           |
|---|-----------|
| <b>1 Code Documentaion:</b>                     | <b>3</b>  |
| 1.1 fabric_package_management package . . . . . | 3         |
| 1.1.1 apt module . . . . .                      | 3         |
| <b>2 Example</b>                                | <b>7</b>  |
| <b>3 Indices and tables</b>                     | <b>9</b>  |
| <b>Python Module Index</b>                      | <b>11</b> |



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.apk.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.apk.source` (*package*, *download\_only=False*, *use\_sudo=False*, *verbose=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.apk.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.apk.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`



---

## Python Module Index

---

### 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