

---

# **Apt Documentation**

***Release 0.3.3***

**Matt Wilkie**

**Mar 25, 2017**



---

## Contents

---

<b>1</b>	<b>Module Reference:</b>	<b>3</b>
1.1	apt Module . . . . .	3
<b>2</b>	<b>Links:</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>



What does apt do? In essence, install applications from [Osgeo4W](#) :

- Download a package from the mirror,
- unpack it under OSGEO4W\_ROOT,
- run any post-install scripts,
- update installed applications list,
- save metadata (cache folder, mirror list).

Show information about installed applications

Remove installed packages



# CHAPTER 1

---

## Module Reference:

---

### apt Module

Apt command line installer and package manager for Osgeo4W.

#### Examples

Typical daily use:

```
apt update           (fetch up-to-date setup.ini)
apt install gdal gdal-python (install packages "gdal" and "gdal-python", and
↳dependencies)
apt new              (show possible upgrades)
apt list             (show installed packages)
apt available        (show installation candidates)
apt remove xxx yyy   (uninstall packages xxx and yyy)
```

#### Notes

Apt strives to match Osgeo4wSetup.exe's results as closely as possible, and uses the same configuration and install manifest files. A prime directive is that user's should never be put in a position where they feel the need to choose between the tools and not go back.

That's the aspiration. There's no guarantee it's been achieved.

At the moment apt can only install the 32bit Osgeo4W packages.

#### References

apt.**available** (*dummy*)  
Show packages available on the mirror.

Display packages available on the mirror, with installed packages marked \*. Specify an alternate mirror with `--mirror=...`

**Parameters** `dummy` (*str*) – Parameter is not used at present.

**Returns** Package names (without install mark).

**Return type** `list`

`apt.ball` (*packages*)

Print full local path name of package archive

C:> apt ball shell

shell = d:/temp/o4w-cache/setup/http%3a%2f%2fdownload.osgeo.org%2fosgeo4w%2f/x86

/release/shell/shell-1.0.0-13.tar.bz2

FIXME: This should either return a list of archive filenames, or there should be a `get_ball(p)` which returns 1 filename, or we should rip out all this repetitive code spread across multiple functions, for the purpose of allowing multiple package input. We need a handler for this instead.

`apt.check_env` (*o4w=''*)

Verify we're running in an Osgeo4W-ready shell

`apt.check_setup` (*installed\_db, setup\_ini*)

Look to see if the installed packages db and setup.ini are available

`apt.datetime_to_unixtime` (*dt, epoch=datetime.datetime(1970, 1, 1, 0, 0)*)

Convert a datetime object to unix UTC time (seconds since beginning).

Adapted from <http://stackoverflow.com/questions/8777753/converting-datetime-date-to-utc-timestamp-in-python/>

It wants `from __future__ import division`, but that caused issues in other functions, automatically converting what used to produce integers into floats (e.g. "50/2"). It seems to be safe to not use it, but leaving this note just in case...

`apt.do_download` (*packagename*)

Download package from mirror and save in local cache folder.

Overwrites existing cached version if md5 sum doesn't match expected from setup.ini.

Returns *path archive.bz2* on success (file downloaded, or file with correct md5 is present), and http status code if fails.

`apt.do_install` (*packagename*)

Unpack the package in appropriate locations, write file list to installed manifest, run postinstall configuration.

`apt.do_run_preremove` (*root, packagename*)

Run the etc/preremove batch files for this package

`apt.do_uninstall` (*packagename*)

For package X: delete installed files & remove from manifest, remove from installed.db

`apt.dodo_download` (*url, dstFile*)

Dumbest name for abstracting downloading a file to disk with requests module and progress reporting

Returns *path archive.bz2* on success, http status code if fails.

`apt.down_stat` (*downloaded\_size, total\_size*)

Report download progress in bar, percent, and bytes.

Each bar stroke '=' is approximately 2%

**Adapted from** <http://stackoverflow.com/questions/51212/how-to-write-a-download-progress-indicator-in-python>  
<http://stackoverflow.com/questions/15644964/python-progress-bar-and-downloads>



`apt.download (packages)`

Download the package(s) from mirror and save in local cache folder:

```
C:> apt download shell gdal {...etc}
```

```
shell = d:/temp/o4w-cache/setup/http%3a%2f%2fdownload.osgeo.org%2fosgeo4w%2f/x86/release/shell/shell-1.0.0-13.tar.bz2    remote:      c38f03d2b7160f891fc36ec776ca4685    shell-1.0.0-13.tar.bz2    local: c38f03d2b7160f891fc36ec776ca4685 shell-1.0.0-13.tar.bz2
```

```
gdal = d:/temp/o4w-cache/setup/http%3a%2f%2fdownload.osgeo.org%2fosgeo4w%2f/x86/release/gdal/gdal-1.11.1-4.tar.bz2    remote:      3b60f036f0d29c401d0927a9ae000f0c    gdal-1.11.1-4.tar.bz2    local: 3b60f036f0d29c401d0927a9ae000f0c gdal-1.11.1-4.tar.bz2
```

Use `apt available` to see what is on the mirror for downloading.

`apt.exceptionHandler (exception_type, exception, traceback, debug_hook=<built-in function exceptionhook>)`

Print user friendly error messages normally, full traceback if DEBUG on. Adapted from <http://stackoverflow.com/questions/27674602/hide-stderr-unless-a-debug-flag-is-set>

`apt.find (patterns)`

Search installed packages for filenames matching the specified text string.

`apt.get_all_dependencies (packages, nested_deps, parent=None)`

Recursive lookup for required packages in order of dependence. Returns an ordered list.

`apt.get_arch (bits)`

DRAFT, unused. Would rather do this because X86\_64 is awkward to type on command line compared to '64' or '64bit'. Need to use `setuprc` first though.

What happens if bitness is not declared? Or set to 64 on one run and then 32 the next? What does mainline setup do? ...I don't know enough.

`apt.get_cache_dir ()`

Return path to use for saving downloads.

#### Precedence order:

- command line option (-c, --cache)
- last used cache (read from `setup.rc`)
- Public Downloads folder
- Osgeo default (%osgeo4w\_root%/var/...)

`apt.get_config (fname)`

Open `/etc/setup/fname` and return contents, e.g. `/etc/setup/last-cache`

`apt.get_filelist (packagename)`

Retrieve list of files installed for package X from manifest (`/etc/setup/package.lst.gz`)

`apt.get_info (packagename)`

Retrieve details for package X.

Returns dict of information for the package from dict created by `parse_setup_ini()` (category, version, archive name, etc.)

`apt.get_installed ()`

Get list of installed packages from `./etc/setup/installed.db`.

Returns nested dictionary (empty when `installed.db` doesn't exist): {status\_int : {pkg\_name : archive\_name}}

I don't know significance of the nesting or leading zero. It appears to be extraneous? The db is just a straight name:tarball lookup table. In `write_installed()` the "status" is hard coded as 0 for all packages.

`apt.get_installed_version(packagename)`  
Derive version number from archive filename in ‘installed’ dict.

`apt.get_menu_links(bat)`  
Parse postinstall batch file for menu and desktop links.  
  
Relies on shlex module which splits on spaces, yet preserves spaces within quotes (<http://stackoverflow.com/questions/79968>)

`apt.get_missing(packagename)`  
For package, identify any requirements (dependencies) that are not installed.  
  
Returns a dictionary of {packagename: ['missing\_1','missing\_2',...']}

`apt.get_new()`  
Return list of mirror packages of newer versions than those installed.

`apt.get_requires(packagename)`  
identify dependencies of package [deprecated]  
  
use `get_all_dependencies()` for recursive dependencies list and `get_info(p)['requires']` for just one level

`apt.get_setup_arch(setup_ini)`  
Return CPU architecture used in setup.ini

`apt.get_special_folder(intFolder)`  
Fetch paths of Windows special folders: Program Files, Desktop, Startmenu, etc.  
  
Written by Luke Pinner, 2010. Code is public domain, do with it what you will... todo: look at replacing with WinShell module by Tim Golden, <http://winshell.readthedocs.org/en/latest/special-folders.html>

`apt.get_zipfile(packagename)`  
Return full path name of locally downloaded package archive.

`apt.hashcheck(package)`  
Check if the md5 hash for “package” in local cache matches mirror  
  
> apt hashcheck shell

**Returns: True or False for md5 match status** None when cache file not found

If passed a list it only processes the first item.

`apt.help(*args)`  
Show help for COMMAND

`apt.info(packages)`  
info - report name, version, category, etc. about the package(s)

B:> apt info shell

name : shell version : 1.0.0-13 sdesc : “OSGeo4W Command Shell” ldesc : “Menu and Desktop icon launch OSGeo4W command shell” category : Commandline\_Uutilities requires : msvcrt setup zip\_path : x86/release/shell/shell-1.0.0-13.tar.bz2 zip\_size : 3763 md5 : c38f03d2b7160f891fc36ec776ca4685 local\_zip: d:/temp/o4w-cache/setup/http%3.../shell-1.0.0-13.tar.bz2 installed: True install\_v: 1.0.0-11

## Notes

- “local\_zip” is best guess based on current mirror. (We don’t record which mirror was in use at the time of package install.)
- “version” is from setup.ini, what is available on the mirror server

•“install\_v” is the version currently installed

`apt.install (packages, force=False)`

Download and install packages, including dependencies

C:> apt install shell gdal

`apt.list_installed (dummy)`

List installed packages

`apt.listfiles (packages)`

List files installed with package X. Multiple packages can be specified.

C:> apt listfiles shell gdal

— shell — OSGeo4W.bat OSGeo4W.ico bin ...etc

— gdal — bin bin/gdal111.dll bin/gdaladdo.exe ...etc

`apt.missing (dummy)`

Display missing dependencies for all installed packages.

*dummy* parameter is ignored

`apt.new (dummy)`

List available upgrades to currently installed packages

`apt.parse_setup_ini (fname)`

Parse setup.ini into package name, description, version, dependencies, etc.

**Parameters** *fname* – full path to setup.ini

**Returns**

```
{Distribution {Program_name{['category', 'source', 'ldesc', 'version', 'install', 'sdesc', 're-
quires']}}}
```

```
{curr {
```

```
  'gdal' { 'name': 'gdal', 'version': '1.11.1-4', 'category': 'Libs Commandline_Uutilities',
    etc... }
```

```
}}
```

**Return type** A nested dictionary

`apt.parse_setuprc (fname)`

Parse setup.rc config file into a dictionary.

We assume any line beginning with a tab is a value, and all others are dict keys. Consequently this will return a bad dict if there are extra lines starting with tabs.

Example C:OSGeo4Wetcsetupsetup.rc:

**mirrors-1st** <http://download.osgeo.org/osgeo4w/OSGeo;USA;California>

**window-placement** 44,0,0,0,0,0,0,1,0,0,0,255,255,255,255,255,255,255...

**last-mode** Advanced

**last-mirror** <http://download.osgeo.org/osgeo4w/>

**net-method** Direct

**last-cache** C:UsersMattDownloads

**last-menu-name** OSGeo4W\_default

And result:

```
last-cache:      C:UsersMattDownloads  last-mirror:      http://download.osgeo.org/osgeo4w/
mirrors-1st:     http://download.osgeo.org/osgeo4w/;OSGeo;USA;Cal...  window-placement:
44,0,0,0,0,0,0,1,0,0,0,255,255,255,255...  last-mode:      Advanced  last-menu-name:   OS-
Geo4W_default net-method: Direct
```

`apt.post_install (packagename)`

**Run postinstall batch files and update package manifest** to catch those files not included in the package archive. (manifest = etc/setup/pkg-foo.lst.gz)

adapted from “17.1.3.3 Replacing os.system()” <http://www.python.org/doc/2.5.2/lib/node536.html>

`apt.remove (packages)`

Uninstall listed packages

`apt.requires (packages)`

What packages does X rely on?

Returns dictionary of package names and dependencies. Reports sub-dependencies, but they aren’t in the dict (yet).

`apt.search (pattern)`

Search available packages list and descriptions for X

Returns list of package names

`apt.set_extended_info (d)`

set extended information into package-info-dictionary, as used by `get_info()` or `parse_setup_ini()`

We take compound values in single keys and explode them into their own keys.

```
{‘install’: ‘x86/release/gdal/gdal-1.11.1-4.tar.bz2 5430991 3b60f036f0d29c401d0927a9ae000f0c’}
```

becomes:

```
{‘zip_path’: ‘x86/release/gdal/gdal-1.11.1-4.tar.bz2’}      {‘zip_size’:‘5430991’}
{‘md5’:‘3b60f036f0d29c401d0927a9ae000f0c’}
```

`apt.setup (target)`

Create skeleton Osgo4W folders and setup database environment

`apt.split_ball (filename)`

Parse package archive name into a) package name and b) version numbers tuple (to feed into `version_to_string`)

mc-4.6.0a-20030721-12.tar.bz2

mc → package name 4.6.0a-20030721 → upstream application version 12 → package version

python-numpy-2.7-1.5.1-1.tar.bz2

python-numpy → package name 2.7-1.5.1 → upstream application version 1 → package version

**Returns** (‘mc’, (4, 6, 0a, 20030721, 12)) (‘python-numpy’, (2, 7, 1, 5, 1, 1))

`apt.uniq (alist)`

Returns a list with unique items (removes duplicates), without losing item order. From @jamyak, <http://stackoverflow.com/a/17016257/14420>

`apt.unique (L)`

Remove duplicates and empty items from a list

`apt.update ()`

Fetch updated package list from mirror.

`apt update`

Specify mirror (web server, windows file share, local disk):

```
apt -mirror=http://example.com/... update apt -mirror=file:///server/share/... update apt -
mirror=file:///D:/downloads/cache/... update
```

`apt.upgrade (packages)`

Upgrade named packages.

`apt upgrade all apt upgrade gdal-filegdb qgis-grass-plugin`

`apt.url (packages)`

Print remote package archive path, relative to mirror root

`apt.url_time_to_datetime (s)`

Convert “last-modified” string time from a web server header to a python datetime object.

Assumes the string looks like “Fri, 27 Mar 2015 08:05:42 GMT”. There is no attempt to use locale or similar, so the function isn’t very robust.

`apt.version (packages)`

Report installed version of X

`apt.write_installed ()`

Record installed packages in install.db

`apt.write_setuprc (setuprc, fname='setup.rc')`

Write the setuprc dictionary to file, in osgeo4w-setup.exe format.

Dict entries with empty values are left out.

**Incoming dict:** last-mode: None last-mirror: <http://download.osgeo.org/osgeo4w/> net-method: None last-cache: C:\Users\Matt\Downloads last-menu-name: OSGeo4W\_default

**Out etc/setup/setup.rc:**

**last-mirror** <http://download.osgeo.org/osgeo4w/>

**last-cache** C:\Users\Matt\Downloads

**last-menu-name** OSGeo4W\_default

`class apt.xAttrDict (*args, **kwargs)`

Access a dictionary by attributes, like using javascript dotted notation.

`dict.mykey` <— same as —> `dict['mykey']`

From <http://stackoverflow.com/questions/4984647/accessing-dict-keys-like-an-attribute-in-python>

`apt.xrequires (packages)`

<https://github.com/maphew/apt/issues/32>



## CHAPTER 2

---

Links:

---

- Issue Tracker: <https://github.com/maphew/apt/issues>
- Source Code: <https://github.com/maphew/apt>
- Docs: <http://apt.readthedocs.org/en/latest/>
- `genindex`





**a**

apt, 3



## A

apt (module), 3  
available() (in module apt), 3

## B

ball() (in module apt), 4

## C

check\_env() (in module apt), 4  
check\_setup() (in module apt), 4

## D

datetime\_to\_unixtime() (in module apt), 4  
do\_download() (in module apt), 4  
do\_install() (in module apt), 4  
do\_run\_preremove() (in module apt), 4  
do\_uninstall() (in module apt), 4  
dodo\_download() (in module apt), 4  
down\_stat() (in module apt), 4  
download() (in module apt), 4

## E

exceptionHandler() (in module apt), 5

## F

find() (in module apt), 5

## G

get\_all\_dependencies() (in module apt), 5  
get\_arch() (in module apt), 5  
get\_cache\_dir() (in module apt), 5  
get\_config() (in module apt), 5  
get\_filelist() (in module apt), 5  
get\_info() (in module apt), 5  
get\_installed() (in module apt), 5  
get\_installed\_version() (in module apt), 5  
get\_menu\_links() (in module apt), 6  
get\_missing() (in module apt), 6  
get\_new() (in module apt), 6

get\_requires() (in module apt), 6  
get\_setup\_arch() (in module apt), 6  
get\_special\_folder() (in module apt), 6  
get\_zipfile() (in module apt), 6

## H

hashcheck() (in module apt), 6  
help() (in module apt), 6

## I

info() (in module apt), 6  
install() (in module apt), 7

## L

list\_installed() (in module apt), 7  
listfiles() (in module apt), 7

## M

missing() (in module apt), 7

## N

new() (in module apt), 7

## P

parse\_setup\_ini() (in module apt), 7  
parse\_setuprc() (in module apt), 7  
post\_install() (in module apt), 8

## R

remove() (in module apt), 8  
requires() (in module apt), 8

## S

search() (in module apt), 8  
set\_extended\_info() (in module apt), 8  
setup() (in module apt), 8  
split\_ball() (in module apt), 8

## U

`uniq()` (in module `apt`), [8](#)  
`unique()` (in module `apt`), [8](#)  
`update()` (in module `apt`), [8](#)  
`upgrade()` (in module `apt`), [9](#)  
`url()` (in module `apt`), [9](#)  
`url_time_to_datetime()` (in module `apt`), [9](#)

## V

`version()` (in module `apt`), [9](#)

## W

`write_installed()` (in module `apt`), [9](#)  
`write_setuprc()` (in module `apt`), [9](#)

## X

`xAttrDict` (class in `apt`), [9](#)  
`xrequires()` (in module `apt`), [9](#)