Some command I often use:

to see dependencies and reverse dependencies:

```
apt-cache [--important] [--installed] [--recurse] depends $package
apt-cache rdepends $package
```

To remove a package, all autoremove packages, and all related config files:

```
sudo apt-get --purge --auto-remove purge $package
```

To see actual apt configuration options:

```
apt-config dump | less
```

To modify an option on the command line

```
$ apt-config dump | grep -i recommend
APT::Install-Recommends "1";
$ sudo apt-get -o APT::Install-Recommends="0" install $package
```

This was just and example of specifying APT options through the command line, to avoid installing recommended packages, you can use:

```
sudo apt-get --no-install-recommends $package
```

To see all local/obsolete packages:

```
aptitude search ~o
```

To see removed packages with residual configuration:

```
aptitude search ~c
```

and to remove them

```
sudo aptitude purge ~c
```

To obtain a list of all installed packages (dpkg -l can sometimes give truncated columns output)

```
dpkg --get-selections | awk '{ print $1 }'
```

or

```
dpkg-query -Wf '${Package}\n'  # other fields available, see man page
```

To dissasemble, modify then riassemble a package

```
dpkg-deb -x file.deb ./dir
cd dir
dpkg-deb -e ../file.deb
# apply your modification, then...
cd ..
dpkg-deb -b dir file-new.deb
```

shareimprove this answer

edited Sep 9 '11 at 17:35

community wiki

4 revs, 2 users 97%

enzotib
add comment
up vote 3 down vote

View the ChangeLog of a package

$ aptitude changelog <pkgname>

Example:

$ aptitude changelog sudo
sudo (1.7.0-1ubuntu2.4) karmic-security; urgency=low

  * SECURITY UPDATE: properly handle multiple PATH variables when using
    secure_path in env.c
    - Adapted http://www.sudo.ws/repos/sudo/raw-rev/a09c6812eaec
    - CVE-2010-1646
...

shareimprove this answer

answered Nov 15 '10 at 6:34

community wiki

GÃ¶del

2

As of Ubuntu 11.04 Natty, apt-get changelog sudo works too. â€“ Lekensteyn Sep 9 '11 at 16:39
add comment
up vote 3 down vote

Two that I use a lot are:-

apt-get autoremove <packagename>

Which will remove the package and any unused dependancies, which is useful if you try an app out, then decide you don't need it, and want the cruft to be removed also.

dpkg -S /path/to/file

Which tells me which package a file was installed with.

Finally, one more..

dpkg -l <packagename> | grep ^ii

Lists packages but only those that have the status ii which means they're installed, so it wont show stuff I've removed.

If you wish to get the package name for a file which was not installed (dpkg -S, but for non-installed packages), install apt-file and run:

apt-file search /path/to/file

shareimprove this answer

edited Sep 9 '11 at 16:41

community wiki

2 revs, 2 users 91%
popey

I find it useful to use which together with dpkg -S. e.g: for executables that live in packages with different names, like: dpkg -S `which uname` â€“ Benjamin Rubin Nov 20 '10 at 0:26

Ooh, yes, that's a good one too. â€“ popey Nov 20 '10 at 8:35
add comment
up vote 2 down vote

Install apt-file, then run sudo apt-file update. You can now search for files in packages that you don't even have installed.

Also handy if you need to know information about packages in other versions of Ubuntu is rmadison, which is in the devscripts package. Provide it with a package name as an argument and it will tell you what versions of that package exist in every current Ubuntu version, and what repository section the package is in.

Example:

```
[bnrubin@server:~/]$ rmadison cowsay
   cowsay |     3.03-8 | dapper/universe | source, all
   cowsay |     3.03-9 | hardy/universe | source, all
   cowsay |   3.03-9.2 | jaunty/universe | source, all
   cowsay |   3.03-9.2 | karmic/universe | source, all
   cowsay |   3.03-9.2 | lucid/universe | source, all
   cowsay | 3.03+dfsg1-2 | maverick/universe | source, all
   cowsay | 3.03+dfsg1-2 | natty/universe | source, all
```

shareimprove this answer

answered Nov 20 '10 at 0:23

community wiki

Benjamin Rubin

apt-file now is able to manage a user's database, so that you do not need to be root to update. â€“ enzotib Sep 9 '11 at 16:49

apt-file is quite nice. thanks for sharing. â€“ GÃ¶del Oct 2 '11 at 8:38
add comment
up vote 1 down vote

To get list commands starts with 'apt-' you do the following. open a terminal and type 'apt-' and press TAB key twice this will list all commands starts with 'apt-'.

Sample output:

```
apt-add-repository    apt-extracttemplates  apt-key
apt-cache             apt-file              apt-mark
apt-cdrom             apt-ftparchive        apt-sortpkgs
apt-config            apt-get
```

To get a detailed information you can check man page of that a specific command

eg: man apt-get
shareimprove this answer

answered Nov 15 '10 at 5:46

community wiki

aneeshep
add comment
up vote 1 down vote

I like to think of three different kind of packages:

- System packages (essential packages or packages of priority standard or higher)
- User packages (manually installed packages of priority optional or extra)
- Dependencies and recommends (automatically installed packages / everything that is not a system or a user package)

To show all "system packages" you can use

aptitude search '(~pstandard|~pimportant|~prequired|~E)'

I like to have all of them installed and marked as manually installed.

aptitude install '(~pstandard|~pimportant|~prequired|~E)!~i'
aptitude unmarkauto '(~pstandard|~pimportant|~prequired|~E)~i~M'

To show all "user packages" use

aptitude search '~i!~M!(~pstandard|~pimportant|~prequired|~E)'

In this list there should be only packages that you know that you want. All other packages are probably just dependencies or recommends of other packages, you can mark them as automatically installed

aptitude markauto libsomething

Take a look at aptitude's Search Term Reference and
What is an Essential, Required, Important, Standard, Optional, or Extra package? for background information.
shareimprove this answer

answered Sep 20 '11 at 12:07

community wiki

Dario
add comment
up vote 0 down vote

Upgrade packages which would be kept back because they would remove other packages or because it's a kernel upgrade:

sudo apt-get dist-upgrade

Purge a package and its config.

sudo apt-get purge package

Show details of a package as known in the package database, including section, version, dependencies, maintainer

and description.

apt-cache show package

List files in an installed package

dpkg -L pkg

Upgrade all packages

sudo apt-get upgrade

shareimprove this answer

edited Sep 9 '11 at 16:47

community wiki

2 revs, 2 users 67%
PinoSan
add comment
up vote -1 down vote

dpkg -i --force-architecture something.i386.deb

For installing some i386 debs on amd64.