Common APT usage

apt-get install ⟨*package*⟩ Downloads ⟨*package*⟩ and all of its dependencies, and installs or upgrades them. This will also take a package off of *hold* if it was put on. See below for more info on *hold.*

apt-get remove [--purge] ⟨*package*⟩ Removes ⟨*package*⟩ and any packages that depend on it. --purge specifies that packages should be *purged*, see `dpkg -P` for more information.

apt-get update Updates packages listings from Debian mirrors, should be run at least once a day if you install anything that day, and every time after /etc/apt/sources.list is changed.

apt-get upgrade [-u] Upgrades all packages installed to newest versions available. Will not install new or remove old packages. If a package changes dependencies and requires installation of a new package, it will not be upgraded, it will be put on *hold* instead. `apt-get upgrade` will not upgrade packages put on *hold* (that is the meaning of *hold*). See below for how to manually put packages on *hold*. I suggest the '-u' option as well, because then you can see what packages are going to be upgraded.

apt-get dist-upgrade [-u] Similar to `apt-get upgrade`, except that *dist-upgrade* will install or remove packages to satisfy dependencies.

apt-cache search ⟨*pattern*⟩ Searches packages and descriptions for ⟨*pattern*⟩.

apt-cache show ⟨*package*⟩ Shows the full description of ⟨*package*⟩.

apt-cache showpkg ⟨*package*⟩ Shows a lot more detail about ⟨*package*⟩, and its relationships to other packages.

synaptic
gdebi
gdebi-kde
dselect
aptitude
gnome-apt Graphical front ends to *APT* (some of these may be in their own package, that must be installed before use). While `dselect` is arguably the most powerful, it's also the oldest and hardest to use.

Common Dpkg usage

dpkg -i ⟨*package.deb*⟩ Installs a Debian package file; one that you downloaded manually, for example.

dpkg -c ⟨*package.deb*⟩ Lists the contents of ⟨*package.deb*⟩, a .deb file.

dpkg -I ⟨*package.deb*⟩ Extracts package information from ⟨*package.deb*⟩, a .deb file.

dpkg -r ⟨*package*⟩ Removes an installed package named ⟨*package*⟩

dpkg -P ⟨*package*⟩ Purges an installed package named ⟨*package*⟩. The difference between *remove* and *purge* is that while *remove* only deletes data and executables, *purge* also deletes all configuration files in addition.

dpkg -L ⟨*package*⟩ Gives a listing of all the files installed by ⟨*package*⟩. See also `dpkg -c` for checking the contents of a .deb file.

dpkg -s ⟨*package*⟩ Shows information on the installed package ⟨*package*⟩. See also `apt-cache show` for viewing package information in the Debian archive and `dpkg -I` for viewing package information extracted from a .deb file.

dpkg-reconfigure ⟨*package*⟩ Reconfigures an installed package, if it uses *debconf* (*debconf* provides that consistent configuration interface for package installation). You can reconfigure *debconf* itself if you want to change the front-end or priority of questions asked. For example, to reconfigure *debconf* with the dialog front-end, you simply run:

> dpkg-reconfigure --frontend=dialog debconf

echo "⟨*package*⟩ hold" | dpkg --set-selections Put ⟨*package*⟩ on *hold* (command line method)

dpkg --get-selections "⟨*package*⟩" Get the current status of ⟨*package*⟩ (command line method)

dpkg -S ⟨*file*⟩ Searches for ⟨*file*⟩ in package database, telling you which installed packages have that file in them. To search for a file or list the contents of a package without installing it, you can use apt-file command line tool.

Building Debian packages from Source

apt-get source [-b] ⟨*package*⟩ Download the source Debian package for ⟨*package*⟩ and extract it. You must have deb-src lines in your /etc/apt/sources.list for this to work. If you supply the '-b' option and you are currently root, then the package will be automatically built if possible.

apt-get build-dep ⟨*package*⟩ Download and install the packages necessary to build the source Debian package ⟨*package*⟩. This feature is only present in `apt` version 0.5 and up. Currently this means that woody and above contain this functionality. If you have an older version of `apt` then the easiest way to find out the build dependencies is to look in the debian/control file in the source package directory. A common usage of this command is in conjunction with `apt-get source -b`. For example (as root):

> apt-get build-dep ⟨*package*⟩
> apt-get source -b ⟨*package*⟩

Will download the source package, all of its build dependencies, and attempt to compile the source package.

dpkg-source -x ⟨*package.dsc*⟩ If you have downloaded the source package for a program manually, which includes several files such as a .orig.tar.gz (or .tar.gz if it is Debian native), a .dsc, and a .diff.gz (if it is not Debian native), then you can unpack the source package using this command on the .dsc file.

dpkg-buildpackage Builds a Debian package from a Debian source tree. You must be in the main directory of the source tree for this to work. Sample usage:

> dpkg-buildpackage -rfakeroot -uc -b

Where '-rfakeroot' instructs it to use the `fakeroot` program to simulate root privileges (for ownership purposes), '-uc' stands for "Don't cryptographically sign the changelog", and '-b' stands for "Build the binary package only"

debuild A handy wrapper script around `dpkg-buildpackage` that will automatically take care of using `fakeroot` or not, as well as running `lintian` and `gpg` for you. This script is provided by (:program), so you need to install this package first.

Fixing dependencies

dpkg --configure --pending If dpkg quits with an error while `apt-get install, upgrade, or dist-upgrade`ing try running this to configure the packages that were already unpacked. Then try `apt-get install, upgrade, or dist-upgrade -f`, and then try `apt-get install, upgrade, or dist-upgrade` again. Repeat as needed. This usually resolves most dependency problems (also, if it mentions a specific package for some reason, you might want to try installing or removing that package)

apt-get install -f
apt-get upgrade -f
apt-get dist-upgrade -f Attempt to fix dependencies while doing one of the above. Note that `apt-get install -f` does not require a ⟨*package*⟩ argument.

See Also

man ⟨***package***⟩ To learn more about this commands you can always consult their man pages, e.g. man apt-get, man dpkg. You can also find information about this and more tools, HOWTOs, manuals, etc at http://www.debian.org/doc/.