8.1.1 dpkg

This is the main package management program. dpkg can be invoked with many options. Some common uses are:

Find out all the options: dpkg --help.

Print out the control file (and other information) for a specified package: dpkg --info foo_VVV-RRR.deb

Install a package (including unpacking and configuring) onto the file system of the hard disk: dpkg --install foo_VVV-RRR.deb.

Unpack (but do not configure) a Debian archive into the file system of the hard disk: dpkg --unpack foo_VVV-RRR.deb. Note that this operation does not necessarily leave the package in a usable state; some files may need further customization to run properly. This command removes any already-installed version of the program and runs the preinst (see What is a Debian preinst, postinst, prerm, and postrm script?, Section 7.6) script associated with the package.

Configure a package that already has been unpacked: dpkg --configure foo. Among other things, this action runs the postinst (see What is a Debian preinst, postinst, prerm, and postrm script?, Section 7.6) script associated with the package. It also updates the files listed in the conffiles for this package. Notice that the 'configure' operation takes as its argument a package name (e.g., foo), not the name of a Debian archive file (e.g., foo_VVV-RRR.deb).

Extract a single file named "blurf" (or a group of files named "blurf*" from a Debian archive: dpkg --fsys-tarfile foo_VVV-RRR.deb | tar -xf - blurf*

Remove a package (but not its configuration files): dpkg --remove foo.

Remove a package (including its configuration files): dpkg --purge foo.

List the installation status of packages containing the string (or regular expression) "foo*": dpkg --list 'foo*'.

8.1.2 APT

APT is the Advanced Package Tool and provides the apt-get program. apt-get provides a simple way to retrieve and install packages from multiple sources using the command line. Unlike dpkg, apt-get does not understand .deb files, it works with the packages proper name and can only install .deb archives from a source specified in /etc/apt/sources.list. apt-get will call dpkg directly after downloading the .deb archives[5] from the configured sources.

Some common ways to use apt-get are:

To update the list of package known by your system, you can run:

```
apt-get update
```

(you should execute this regularly to update your package lists)

To upgrade all the packages on your system (without installing extra packages or removing packages), run:

```
apt-get upgrade
```

To install the foo package and all its dependencies, run:

```
apt-get install foo
```

To remove the foo package from your system, run:

```
apt-get remove foo
```

To remove the foo package and its configuration files from your system, run:

```
apt-get --purge remove foo
```

To upgrade all the packages on your system, and, if needed for a package upgrade, installing extra packages or removing packages, run:

```
apt-get dist-upgrade
```

(The command upgrade keeps a package at its installed obsolete version if upgrading would need an extra package to be installed, for a new dependency to be satisfied. The dist-upgrade command is less conservative.)

Note that you must be logged in as root to perform any commands that modify the system packages.

Note that apt-get now installs recommended packages as default and is the preferred program for package management from console to perform system installation and major system upgrades for its robustness.

The apt tool suite also includes the apt-cache tool to query the package lists. You can use it to find packages providing specific functionality through simple text or regular expression queries and through queries of dependencies in the package management system. Some common ways to use apt-cache are:

To find packages whose description contain word:

```
apt-cache search word
```

To print the detailed information of a package:

```
apt-cache show package
```

To print the packages a given package depends on:

```
apt-cache depends package
```

To print detailed information of the versions available for a package and the packages that reverse-depends on it:

```
apt-cache showpkg package
```

For more information, install the apt package and read apt-get(8), sources.list(5) and install the apt-doc package and read /usr/share/doc/apt-doc/guide.html/index.html.

8.1.3 aptitude

aptitude is a package manager for Debian GNU/Linux systems that provides a frontend to the apt package management infrastructure. aptitude is a text-based interface using the curses library, it can be used to perform management tasks in a fast and easy way.

aptitude provides the functionality of dselect and apt-get, as well as many additional features not found in either program:

aptitude offers easy access to all versions of a package.

aptitude makes it easy to keep track of obsolete software by listing it under "Obsolete and Locally Created Packages".

aptitude includes a fairly powerful system for searching particular packages and limiting the package display. Users familiar with mutt will pick up quickly, as mutt was the inspiration for the expression syntax.

aptitude can be used to install the predefined tasks available. For more information see tasksel, Section 8.1.5.

aptitude in full screen mode has su functionality embedded and can be run by a normal user. It will call su (and ask for the root password, if any) when you really need administrative privileges

You can use aptitude through a visual interface (simply run aptitude) or directly from the command line. The command line syntax used is very similar to the one used in apt-get. For example, to install the foo package, you can run aptitude install foo.

Note that aptitude is the preferred program for daily package management from console.

For more informations, read the manual page aptitude(8) and install the aptitude-doc package.

8.1.4 synaptic

synaptic is a graphical package manager. It enables you to install, upgrade and remove software packages in a user friendly way. Next to all features offered by aptitude, it also has a feature for editing the list of used repositories, and supports browsing all available documentation related to a package. See the Synaptic Website for more information.

8.1.5 tasksel

When you want to perform a specific task it might be difficult to find the appropiate suite of packages that fill your need. The Debian developers have defined tasks, a task is a collection of several individual Debian packages all related

to a specific activity. Tasks can be installed through the tasksel program or through aptitude.

The Debian installer will typically install automaticaly the task associated with a standard system and a desktop environment. The specific desktop environment installed will depend on the CD/DVD media used, most commonly it will be the GNOME desktop (gnome-desktop task). Also, depending on your selections throughout the installation process, tasks might be automatically installed in your system. For example, if you selected a language, the task associated with it will be installed automatically too and if you are running in a laptop system the installer recognises the laptop task will be installed too.

8.1.6 Other package management tools

8.1.6.1 dselect

This program is a menu-driven interface to the Debian package management system. For woody and earlier releases, this was the main package management interface for for first-time installations, but currently users are encouraged to use aptitude instead. Some users might feel more comfortable using aptitude and it is also recommended over dselect for large-scale upgrades. For more information on aptitude please see aptitude, Section 8.1.3.

dselect can:

guide the user as he/she chooses among packages to install or remove, ensuring that no packages are installed that conflict with one another, and that all packages required to make each package work properly are installed;

warn the user about inconsistencies or incompatibilities in their selections;

determine the order in which the packages must be installed;

automatically perform the installation or removal; and

guide the user through whatever configuration process are required for each package.

dselect begins by presenting the user with a menu of 7 items, each of which is a specific action. The user can select one of the actions by using the arrow keys to move the highlighter bar, then pressing the <enter> key to select the highlighted action.

What the user sees next depends on the action he selected. If he selects any option but Access or Select, then dselect will simply proceed to execute the specified action: e.g., if the user selected the action Remove, then dselect would proceed to remove all of the files selected for removal when the user last chose the Select action.

Both the Access menu item and the Select menu item lead to additional menus. In both cases, the menus are presented as split screens; the top screen gives a scrollable list of choices, while the bottom screen gives a brief explanation ("info") for each choice.

Extensive on-line help is available, use the '?' key to get to a help screen at any time.

The order in which the actions are presented in the first dselect menu represents the order in which a user would normally choose dselect to install packages. However, a user can pick any of the main menu choices as often as needed (including not at all, depending on what one wants to do).

Begin by choosing an Access Method. This is the method by which the user plans on accessing Debian packages; e.g., some users have Debian packages available on CD-ROM, while others plan to fetch them using anonymous FTP. The selected "Access Method" is stored after dselect exits, so if it does not change, then this option need not be invoked again.

Then Update the list of available packages. To do this, dselect reads the file "Packages.gz" which should be included in the top level of the directory where the Debian packages to be installed are stored. (But if it is not there, dselect will offer to make it for you.)

Select specific packages for installation on his system. After choosing this menu item, the user is first presented with a full screen of help (unless the `--expert' command line option was used). Once the user exits the Help screen, he sees the split-screen menu for choosing packages to install (or remove).

The top part of the screen is a relatively narrow window into the list of Debian's 37400 packages; the bottom part of the

screen contains description of the package or group of packages which are highlighted above.

One can specify which packages should be operated on by highlighting a package name or the label for a group of packages. After that, you can select packages:

to be installed:
This is accomplished by pressing the `+' key.

to be deleted:
Packages can be deleted two ways:

removed: this removes most of the files associated with the package, but preserves the files listed as configuration files (see What is a Debian conffile?, Section 7.5) and package configuration information. This is done by pressing the `-' key.

purged: this removes every file that is part of the package. This is done by pressing the `_' key.

Note that it's not possible to remove "All Packages". If you try that, your system will instead be reduced to the initial installed base packages.

to be put "on hold"
This is done by pressing `=', and it effectively tells dselect not to upgrade a package even if the version currently installed on your system is not as recent as the version that is available in the Debian repository you are using (this was specified when you set the Access Method, and acquired when you used Update).

Just like you can put a package on hold, you can reverse such setting by pressing `:'. That tells dselect that the package(s) may be upgraded if a newer version is available. This is the default setting.

You can select a different order in which the packages are presented, by using the `o' key to cycle between various options for sorting the packages. The default order is to present packages by Priority; within each priority, packages are presented in order of the directory (a.k.a. section) of the archive in which they are stored. Given this sort order, some packages in section A (say) may be presented first, followed by some packages in section B, followed by more packages (of lower priority) in section A.

You can also expand meanings of the labels at the top of the screen, by using the `v' (verbose) key. This action pushes much of the text that formerly fit onto the display off to the right. To see it, press the right arrow; to scroll back to the left, press the left arrow.

If you select a package for installation or removal, e.g., foo.deb, and that package depends on (or recommends) another package, e.g., blurf.deb, then dselect will place the you in a sub-screen of the main selection screen. There you can choose among the related packages, accepting the suggested actions (to install or not), or rejecting them. To do the latter, press Shift-D; to return to the former, press Shift-U. In any case, you can save your selections and return to the main selection screen by pressing Shift-Q.

Users returning to the main menu can then select the "Install" menu item to unpack and configure the selected packages. Alternatively, users wishing to remove files can choose the "Remove" menu item. At any point, users can choose "Quit" to exit dselect; users' selections are preserved by dselect.

8.1.6.2 dpkg-deb

This program manipulates Debian archive(.deb) files. Some common uses are:

Find out all the options: dpkg-deb --help.

Determine what files are contained in a Debian archive file: dpkg-deb --contents foo_VVV-RRR.deb)

Extract the files contained in a named Debian archive into a user specified directory: dpkg-deb --extract foo_VVV-RRR.deb tmp extracts each of the files in foo_VVV-RRR.deb into the directory tmp/. This is convenient for examining the contents of a package in a localized directory, without installing the package into the root file system.

Extract the control information files from a package: dpkg-deb --control foo_VVV-RRR.deb tmp.

Note that any packages that were merely unpacked using dpkg-deb --extract will be incorrectly installed, you should use

dpkg --install instead.

More information is given in the manual page dpkg-deb(1).

8.1.6.3 dpkg-split

This program splits large package into smaller files (e.g., for writing onto a set of floppy disks), and can also be used to merge a set of split files back into a single file. It can only be used on a Debian system (i.e. a system containing the dpkg package), since it calls the program dpkg-deb to parse the debian package file into its component records.

For example, to split a big .deb file into N parts,

Execute the command dpkg-split --split foo.deb. This will produce N files each of approximately 460 KBytes long in the current directory.

Copy those N files to floppy disks.

Copy the contents of the floppy disks onto the hard disk of your choice on the other machine.

Join those part-files together using dpkg-split --join "foo*".

8.2 Debian claims to be able to update a running program; how is this accomplished?

The kernel (file system) in Debian GNU/Linux systems supports replacing files even while they're being used.

We also provide a program called start-stop-daemon which is used to start daemons at boot time or to stop daemons when the runlevel is changed (e.g., from multi-user to single-user or to halt). The same program is used by installation scripts when a new package containing a daemon is installed, to stop running daemons, and restart them as necessary.

8.3 How can I tell what packages are already installed on a Debian system?

To learn the status of all the packages installed on a Debian system, execute the command

```
dpkg --list
```

This prints out a one-line summary for each package, giving a 2-letter status symbol (explained in the header), the package name, the version which is installed, and a brief description.

To learn the status of packages whose names match the string any pattern beginning with "foo" by executing the command:

```
dpkg --list 'foo*'
```

To get a more verbose report for a particular package, execute the command:

```
dpkg --status packagename
```

8.4 How to display the files of a package installed?

To list all the files provided by the installed package foo execute the command

```
dpkg --listfiles foo
```

Note that the files created by the installation scripts aren't displayed.

8.5 How can I find out what package produced a particular file?

To identify the package that produced the file named foo execute either:

dpkg --search filename

This searches for filename in installed packages. (This is (currently) equivalent to searching all of the files having the file extension of .list in the directory /var/lib/dpkg/info/, and adjusting the output to print the names of all the packages containing it, and diversions.)

A faster alternative to this is the dlocate tool.

dlocate -S filename
zgrep foo Contents-ARCH.gz

This searches for files which contain the substring foo in their full path names. The files Contents-ARCH.gz (where ARCH represents the wanted architecture) reside in the major package directories (main, non-free, contrib) at a Debian FTP site (i.e. under /debian/dists/wheezy). A Contents file refers only to the packages in the subdirectory tree where it resides. Therefore, a user might have to search more than one Contents files to find the package containing the file foo.

This method has the advantage over dpkg --search in that it will find files in packages that are not currently installed on your system.

apt-file search foo

If you install the apt-file, similar to the above, it searches files which contain the substring or regular expression foo in their full path names. The advantage over the sample above is that there is no need to retrieve the Contents-ARCH.gz files as it will do this automatically for all the sources defined in /etc/apt/sources.list when you run (as root) apt-file update.

8.6 Why doesn't get `foo-data' removed when I uninstall `foo'? How do I make sure old unused library-packages get purged?

Some packages are split in program (`foo') and data (`foo-data') (or in `foo' and `foo-doc'). This is true for many games, multimedia applications and dictionaries in Debian and has been introduced since some users might want to access the raw data without installing the program or because the program can be run without the data itself, making it optional.

Similar situations occur when dealing with libraries: generally these get installed since packages containing applications depend on them. When the application-package is purged, the library-package might stay on the system. Or: when the application-package no longer depends upon e.g. libdb4.2, but upon libdb4.3, the libdb4.2 package might stay when the application-package is upgraded.

In these cases, `foo-data' doesn't depend on `foo', so when you remove the `foo' package it will not get automatically removed by most package management tools. The same holds true for the library packages. This is necessary to avoid circular dependencies. If you use aptitude (see aptitude, Section 8.1.3) as your package management tool it will, however, track automatically installed packages and remove them when no packages remain that need them in your system

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

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