Installing software on Debian

Small Package Service

In its current version, Debian includes about 9,000 packages. The next version will include around 14,000. Aptitude provides a clearer view and takes the pain out of finding software.

BY DIRK MATERLIK


UPS

Advanced package management with dependency and conflict handling is one of the biggest advantages of Debian **GNU**/Linux [1]. Considering the sheer numbers of packages that Debian includes, most users could use some help finding their way through the masses.

Aptitude provides a GUI that allows users to locate, install, and delete packages. Compared to the command line based install, Aptitude gives a better overview of the package list.

Debian

Debian packages are recognized by the *.deb* extension. They contain a group of related programs and data. Also, and this is where they differ from normal archive files, a package has **metadata**, such as details on the version, and dependencies on other packages.

There is a **maintainer** for each Debian package, who is responsible for packaging the software in a way that allows users to add it to a running Debian system, in line with the guidelines [3].

It is uncommon to handle Debian packages manually. The *dpkg* takes care of this, installing packages or displaying their metadata.

However, dpkg only knows about the software installed on your machine. If a new package requires programs that you have not yet installed, it will issue an error message and quit, leaving it up to the user to resolve the issue. The *APT* ("Advanced Package Tool") program library has the metadata for all the packages in your configured sources, it recognizes dependencies, and resolves conflicts (see Box 1: Debian Sources).

The standard software for APT is a toolkit comprising the command line tools *apt-get*, *apt-cache*, *apt-config*, *apt-cdrom*, and *apt-setup*. The *apt-get install aptitude* installs Aptitude including the required libraries. For a detailed discussion of configuring APT and using the command line tools, check out [4].

Figure 1 shows the toolkit: APT fetches the requested Debian packages

Box 1: Debian Sources

APT parses the configuration file, */etc/apt/sources.list*, to find out where to look for installable packages. It uses a database with all the available packages, and their locations, to do so. If you add a data source, APT will parse the new package list, and add the new programs and versions to its database. Whenever a user wants to install a package, APT will know which CD the user needs to insert, or from which server it can download the package. If the idea of manually editing the *sources.list* file worries you, you can use the *apt-setup*, or *apt-cdrom add* to setup a CD source.

Box 2: Debian Distributions

There are three different Debian distributions. *Stable* is the official release version, and full updates are extremely rare. On the upside, the software in this distribution is fully tested, and security issues are resolved quickly. The current stable version is codenamed *Woody*.

The *unstable* version, which is always known as *Sid*, is where development work takes place. It includes the latest versions of all packages, although they are in a continual state of editing and testing.

Testing is a compromise between the stable and unstable variants. Packages are moved to *Testing* after running for a while in *unstable* without any serious hitches. On the downside, no exceptions to this rule are made for security updates. It can often take a while for a fix to come through. *Testing* is the designated successor of the current *stable* version, and is codenamed *Sarge* at present.

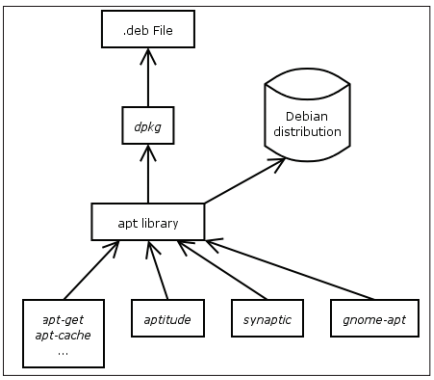

Figure 1: Various layers take the pain out of package installation.

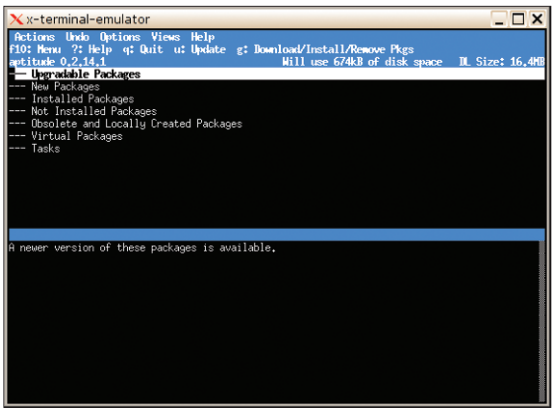

Figure 2: The categories in the default view.

and dependencies from the configured sources, and passes them on to dpkg for installation. Besides the command line tools, and Aptitude, there are a number of GUI-based front-ends including Synaptic [5], KPackage [6], and Gnome-APT.

Front-Ends

Aptitude is an alternative to the command line. It does not provide a GUI-based front-end in the true sense of the word, but instead uses the *curses* library, which works with block characters to draw graphical elements, such as windows and menus. This has the advantage of running on systems that do not have a graphical desktop, or even in a command line based **ssh** session.

Although Aptitude's advantage over apt-get and consorts lies in the interactive user interface, Aptitude can be controlled via the command line. Table 1 has a list of the most important commands. To use them, launch Aptitude as follows:

```
aptitude command
```

When you enter *aptitude* without specifying any parameters, the program initially displays a menu bar, help texts for the most important keys, the package list at the top of the main window, and a description panel with details on the selected entry below (see Figure 2). If a mouse cursor is available, as is the case for a terminal window on a GUI, you can use the mouse to control Aptitude. Otherwise, [F10] will take you to the menu.

Aptitude uses various views, which you can manage by accessing the menu of the same name. The [q] key closes a view. The program quits when you close the last view. [F6] and [F7] take you to the next and previous view, respectively.

-- next to an entry in the package list indicates a category, which you can expand by pressing [Enter] or double-clicking. You can navigate the categories to select an individual package and display details on the package in the description panel (see Figure 3). You can use the [a] and [z] keys to move up and down in the description panel. Package descriptions for Debian are available in English only.

The letter or letters next to the package name indicate the status of the package. The first letter, which is always present, tells you if a package is installed (*i*) or not (*p*). Table 2 indicates the less common alternatives. The second letter tells you the action that Aptitude needs to perform for this package. This is typically *d* for deinstall, or *i* for install. Table 3 shows you the other options. This field is empty if there is no action to perform.

You can press [Enter], or double-click a package, to open up a new view that shows the metadata for the package. The important details here are the dependencies below *Depends*, packages that cannot be installed simultaneously below *Conflicts* and the packages that are not absolutely necessary, but recommended, below *Recommends*. [q] closes and returns to the previous view.

Table 1: Command line actions for Aptitude

Command	Action
install packagename	Installs the specified package.
remove packagename	Uninstalls the specified package.
purge packagename	Uninstalls the specified package and deletes the configuration files.
hold packagename	Prevents updates to the specified package.
unhold packagename	Re-enables updates for the specified package.
markauto packagename	Tags the specified package as automatically installed.
unmarkauto packagename	Tags the specified package as automatically manually installed.
update	Updates the list of installable packages.
upgrade	Updates packages where newer versions are available.
dist-upgrade	Updates packages despite the fact that other packages need to be installed or deleted to do so.
search search pattern	Searches for packages with the search pattern.
clean	Deletes all downloaded package data from disk.
autoclean	Deletes downloaded packages from disk, if a newer version exists.
download packagename	Downloads the specified package without installing it.

GLOSSARY

GNU: *The GNU project ("Gnu is Not Unix") was founded in 1984 with the aim of developing a Unix system based entirely on free software. Linux is the most widespread free Unix-style kernel, but theoretically the Debian project is open for any free alternative, such as the still incomplete Hurd [2] as in the "Debian GNU/Hurd" project. The official name, "Debian GNU/Linux", refers to the Debian GNU system with the Linux kernel.*
Maintainer: *A person who ensures that a software package remains useful and usable.*
ssh: *The "secure shell" allows you to access the command line of another machine across an encrypted connection. You can also use ssh to encrypt more or less any network connection.*
Metadata: *Data on data. For example, a MP3 file contains information on the artist or genre.*

Action!

You can press the [+] key to install a package. This assumes that you have selected the package with the cursor. The current view – this can be the package list or detail view – has no effect on this. Doing so will install the current package and any packages required for this install. The [-] uninstalls a package. To uninstall both the program files and the complete configuration, press [_] instead.

Aptitude remembers the packages it has selected for installing, in order to resolve dependencies for other packages. When the last piece of software that requires a specific package is removed from the system, Aptitude selects it for uninstalling. As this function is provided by Aptitude, and not the APT library, this only works for packages installed by Aptitude.

If you manually select a package to install by pressing the [+] key, nothing happens at first. You need to press [g] for "go" to start the install. Aptitude then displays a new overview of the packages to install and uninstall, sorted by motivation; that is, it indicates whether a package was automatically selected by Aptitude in order to resolve some dependency, or if the user selected the package. You need to press [g] once more to confirm and implement the changes. If you launched Aptitude without the required *root* privileges, the program prompts you for the password (see Figure 4).

If the APT configuration includes Internet servers as data sources, you can press the [u] key to update the lists on these servers. To select installed packages for updating, if a new version is available, go on to press [U]. If you want to prevent a package update, press [=] to keep the current version and ignore any updates. However, if it all somehow goes wrong, you can always press [L] to reinstall and thus repair damaged packages.

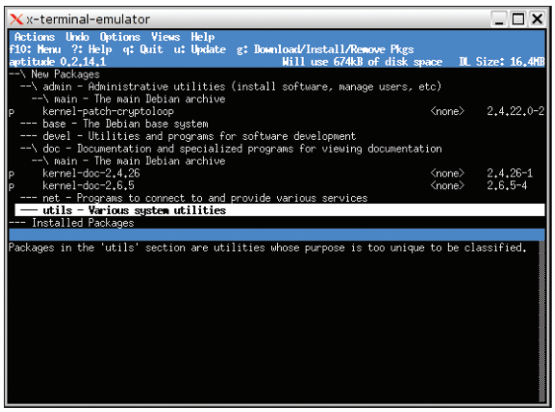

Figure 3: The package list is organized in a hierarchy.

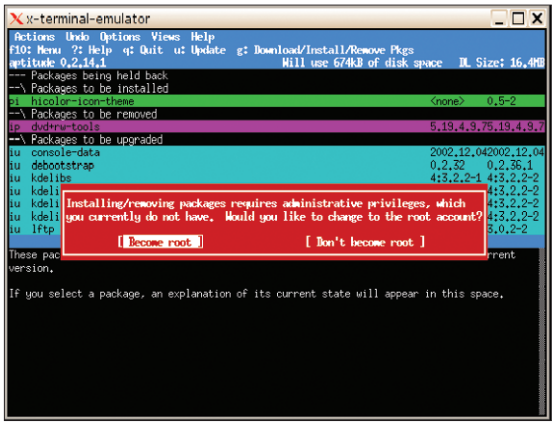

Figure 4: Root privileges are required to complete actions.

Groups

You can define the criteria that Aptitude applies to organize packages in groups. Press [G] to pop up a dialog where you can change the sorting order. The default is as follows:

```
filter(missing),task,➥
status,section(subdir,➥
passthrough),section(topdir)
```

The *filter(missing)* entry hides any non-installable packages. In other words, this option ensures that you can only see packages that are available on a CD or server.

The next entry, *task*, displays the *Task* category. This category contains special packages that do not have any program files of their own, but install other packages because of the dependencies they include. As an example, the *kde* package requires the KDE basic packages. APT automatically installs the packages for the KDE basic system when you select the *kde* package.

The two *section* entries in the default setting, control the groups for the two sections that each Debian package must belong to. The top section (*topdir*) includes *main*, that is the main distribution packages, and *non-free* includes software that Debian categorizes as non-free. At the next structure level, there are *subdir*s, such as *devel* for developer tools, or *games* for games.

The *section(subdir,passthrough)* entry organizes the list by subsection. This places them in front of *section(topdir)*, and means that Aptitude as shown in Figure 3 – will display the top sections in the list of subsections. The *passthrough* option prevents multiple occurrences of packages and categories in more complex structures. If you change the default order of the *section* entries, the order of the entries in the tree changes from *Package status | topdir | subdir* to *Package status | subdir | topdir*.

In addition, you can opt to group packages by *priority* or by the *action* to be

Table 2: Status letter

Letter	Status
i	Installed.
p	Not installed.
v	Virtual: this is not a genuine package. Several genuine packages can be installed to resolve this dependency.
u	Downloaded and unpacked, but not configured.
C	Partly configured.
H	Partly installed.
c	Not installed, but the configuration files are still available – this typically comes from a previous installation.
B	Bad: the status of this package on the system is not unknown.
E	Internal Aptitude error.

Table 3: Action letters

Letter	Action
i	Install.
d	Uninstall: keep the configuration files on the system.
h	Hold: Do not update this package.
p	Delete completely, including the configuration files.
r	Re-install a previously installed package.
u	Update the package to a more recent version.

GLOSSARY

Regular expression: *An approach to formulating searches that is typical of Unix. A period indicates any character, an asterisk any number of repetitions of the previous character. Thus, ".*" will find anything, as it indicates any character and any number of repetitions.*

performed for the package. The latter option will remove entries that do not define an action from the package list.

At the bottom of the list, you can use either the *versions* or *deps* category. This allows you to create your own categories for individual packages. *versions* assigns entries for each version of the package to these new levels, *deps* additionally lists the dependencies and conflicts.

If you simply need a full list of packages, just give an empty string as grouping specifier. *status,action* gives you an overview of the actions due to be performed. [S] sets the order in which Aptitude will display packages within the individual categories. Your options are *name* – the default – for alphabetical order; *installsize*, to sort by required disk space; and *priority*, which indicates the importance of the package. You can use a ~ prefix to reverse the sorting order.

Table 4: Special search options

Option	Action
~b	Search for bad packages.
~d	Search package descriptions.
~i	Search in installed packages.
~m	Search in the *Maintainer* field.
~n	Search in package names.
~s	Search in section names.

Searching for Packages

To find packages that contain a specific string, type [/] to enable the Aptitude search function. Enter a search key in the dialog box that appears to select the first package that matches the search key. [n] repeats the search and selects the next match. Unless otherwise specified, Aptitude will use **regular expressions** to search the packages. This allows for precise searching. For example, *^lib.*gtk* searches for package names that start with *lib* and also include the *gtk* string.

The search can be extended to the other metadata. The *~s* prefix will search for a section. Table 4 shows the possible options.

To avoid the need to navigate the package list if the search returns a large number of hits, you can press [l] to create a view that includes only the matches. In this case, the list displays only packages that match the search key.

If you are accustomed to working with *apt-get* and the accompanying tools, you may be wary of moving to Aptitude. However, Aptitude has features that are difficult or even impossible to provide. The developers are working hard on additional developments, so you can look forward to new features in future. ■

INFO

[1] Debian GNU/Linux: *http://www.debian.org/index.html*

[2] Hurd: *http://www.gnu.org/software/hurd/*

[3] Debian Social Contract and Free Software Guidelines: *http://www.debian.org/social_contract.html*

[4] APT manual: *http://www.debian.org/doc/manuals/apt-howto/index.html*

[5] Synaptic: *http://www.nongnu.org/synaptic/*

[6] KPackage: *http://www.general.uwa.edu.au/u/toivo/kpackage/*