**Package Management**

**Ubuntu features a comprehensive package management system for installing, upgrading, configuring, and removing software.**

**Packages:**

The package files contain all of the necessary files, meta-data, and instructions to implement a particular functionality or software application on your Ubuntu computer.

**dependencies:**

Many complex packages use dependencies. Dependencies are additional packages required by the principal package in order to function properly.

**package managers:**

**1)dpkg:-**

dpkg is a package manager for Debian-based systems. It can install, remove, and build packages, but unlike other package management systems, it cannot automatically download and install packages or their dependencies**.**

You can install a local .deb file by entering:

sudo dpkg -i zip\_3.0-4\_i386.deb

Change zip\_3.0-4\_i386.deb to the actual file name of the local .deb file you wish to install.

remove:

sudo dpkg -r zip

Some more commands for dpkg:

dpkg -c pkg.deb <to view contents of the package>

dpkg -s pkgname <to check if packages are installed or not>

dpkg -l <to check all the installed packages>

**2)Apt-get:**

The apt-get command is a powerful command-line tool, which works with Ubuntu's Advanced Packaging Tool (APT) performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system.

sudo apt-get install,remove,purge etc commands are used to manage packages.

**Update** the Package Index: The APT package index is essentially a database of available packages from the repositories defined in the /etc/apt/sources.list file and in the /etc/apt/sources.list.d directory. To update the local package index with the latest changes made in the repositories, type the following:

sudo apt-get update

**Upgrade** Packages: Over time, updated versions of packages currently installed on your computer may become available from the package repositories (for example security updates). To upgrade your system, first update your package index as outlined above, and then type:

sudo apt-get upgrade

**3)Aptitude:**

Launching Aptitude with no command-line options, will give you a menu-driven, text-based front-end to the Advanced Packaging Tool (APT) system. Many of the common package management functions, such as installation, removal, and upgrade, can be performed in Aptitude with single-key commands, which are typically lowercase letters.

you can start it by typing :

**sudo aptitude**

**Install Packages:** To install a package, locate the package via the Not Installed Packages package category, by using the keyboard arrow keys and the ENTER key. Highlight the desired package, then press the + key. The package entry should turn green, indicating that it has been marked for installation. Now press g to be presented with a summary of package actions. Press g again, and downloading and installation of the package will commence. When finished, press ENTER, to return to the menu.

**Remove Packages:** To remove a package, locate the package via the Installed Packages package category, by using the keyboard arrow keys and the ENTER key. Highlight the desired package you wish to remove, then press the - key. The package entry should turn pink, indicating it has been marked for removal. Now press g to be presented with a summary of package actions. Press g again, and removal of the package will commence. When finished, press ENTER, to return to the menu.

**Update Package Index:** To update the package index, simply press the u key. Updating of the package index will commence.

**Upgrade Packages:** To upgrade packages, perform the update of the package index as detailed above, and then press the U key to mark all packages with updates. Now press g whereby you'll be presented with a summary of package actions. Press g again, and the download and installation will commence. When finished, press ENTER, to return to the menu.

**Command Line Aptitude**

You can also use Aptitude as a command-line tool, similar to apt-get. To install the nmap package with all necessary dependencies, as in the apt-get example, you would use the following command:

sudo aptitude install nmap

To remove the same package, you would use the command:

sudo aptitude remove nmap

Some more commands for aptitude :

aptitude install pkgname(eg. java)

aptitude show pkgname <package info>

aptitude search ~i|head <gives all installed packages>

aptitude search ~b|head <gives broken packages>

aptitude remove pkgname<to remove packages>

aptitude update <updates package repositories>

**4)Synaptic package manager:**

Synaptic is a graphical package management tool based on GTK+ and APT. Synaptic enables you to install, upgrade and remove software packages in a user friendly way.

Besides these basic functions the following features are provided:

* Search and filter the list of available packages
* Perform smart system upgrades
* Fix broken package dependencies
* Edit the list of used repositories (sources.list)
* Download the latest changelog of a package
* Configure packages through the debconf system
* Browse all available documentation related to a package (dwww is required)

**Package states**

1) not-installed

The package is not installed on your system.

2)config-files

Only the configuration files of the package exist on the system.

3)half-installed

The installation of the package has been started, but not

completed for some reason.

4)unpacked

The package is unpacked, but not configured.

5)half-configured

The package is unpacked and configuration has been started, but

not yet completed for some reason.

6)triggers-awaited

The package awaits trigger processing by another package.

7)triggers-pending

The package has been triggered.

8)installed

The package is correctly unpacked and configured.

**unpacking :**

The .deb file contains files to be unpacked in specific locations in the filesystem, including configuration files, and some scripts to be executed before and after installation, upgrading and removal of the package. Failure to run the appropriate steps will result in a borked installation, which will be really hard to revert if the files are not under package manager control. Also, running the scripts may require some environment that's set up by the package manager and that may be quite difficult to replicate by hand.

That said, you can unpack a .deb file by hand:

ar x file.deb

this will leave three files: control.tar.gz, data.tar.gz and debian-binary. The two tarballs contain the control files, checksums and scripts (control.tar.gz) and the actual files to be unpacked on the filesystem (data.tar.gz). You may be able to get what you need by unpacking and analyzing these files.

Every .deb package contains the two files control.tar.gz and data.tar.gz.

data.tar.gz contains all the program's files.

control.tar.gz contains the metadata and some scripts: preinst is run by dpkg before unpacking data.tar.gz, postinst after unpacking. When removing a packge prerm is run before the program files are removed and postrm after that.

**Mostly, apt-get does the following things:**

1)checks for dependencies (and asks to install them),

2)downloads the package, verifies it and then tells dpkg to install it.

**dpkg will:**

1)extract the package and copy the content to the right location, and check for pre-existing files and modifications on them,

2)run package maintainer scripts: preinst, postinst, (and prerm, postrm before these, if a package is being upgraded)

You might be interested in the maintainer scripts, which are usually located at /var/lib/dpkg/info/<package-name>.{pre,post}{rm,inst}. These are usually shell scripts, but there's no hard rule. For example:

$ ls /var/lib/dpkg/info/xml-core.{pre,post}{rm,inst}

/var/lib/dpkg/info/xml-core.postinst

/var/lib/dpkg/info/xml-core.postrm

/var/lib/dpkg/info/xml-core.preinst

/var/lib/dpkg/info/xml-core.prerm

**sources.list**

Configuration of the Advanced Packaging Tool (APT) system repositories is stored in the /etc/apt/sources.list file and the /etc/apt/sources.list.d directory.

**The downloaded packages are stored in /var/cache/apt/archives**

**sudo apt-get clean:**

apt-get clean will not harm your system. The .deb packages in /var/cache/apt/archives are used by the system to install software. But once the software is installed, these .deb packages are no longer needed. You will need these .deb files only while re-installing the software. In case you do not need to re-install, remove these .deb files using apt-get clean.

**remove and purge:**

remove - Packages installed are removed (Does NOT include configuration files)

purge - Purge is identical to remove except that packages are removed and purged. Purge meaning that any configuration files are deleted too.

apt-get --purge remove {package\_name}.

This of course, does not apply to packages that hold configuration files inside the user's home folder (eg: /home/User), this files will not be touched .

So for example, if you were to remove Chrome, Firefox, XBMC or any other that holds some configuration files inside your /home folder, this files will stay there.

On the other hand if you were to install apache, squid, mysql or any other services similar that save their files in /etc, this configuration files will be deleted if you use purge.