Package management

Ubuntu features a comprehensive package management system for installing, upgrading, configuring, and removing software. In addition to providing access to an organized base of over 60,000 software packages for your Ubuntu computer, the package management facilities also feature dependency resolution capabilities and software update checking.

Several tools are available for interacting with Ubuntu’s package management system, from simple command-line utilities that can be easily automated by system administrators, to an easy-to-use graphical interface for those new to Ubuntu.

Introduction

Ubuntu’s package management system is derived from the same system used by the Debian GNU/Linux distribution. The package files contain all of the necessary files, metadata, and instructions to implement a particular functionality or software application on your Ubuntu computer.

Debian package files typically have the extension .deb, and usually exist in *repositories* which are collections of packages found online or on physical media, such as CD-ROM discs. Packages are normally in a pre-compiled binary format; thus installation is quick and requires no compiling of software.

Many packages use *dependencies*. Dependencies are additional packages required by the principal package in order to function properly. For example, the speech synthesis package festival depends upon the package alsa-utils, which is a package supplying the [Advanced Linux Sound Architecture (ALSA)](https://www.alsa-project.org/wiki/Main_Page) sound library tools needed for audio playback. In order for festival to function, it – and all of its dependencies – must be installed. The software management tools in Ubuntu will do this automatically.

Advanced Packaging Tool – APT

The apt command is a powerful command-line tool, which works with Ubuntu’s Advanced Packaging Tool (APT). The commands contained within apt provide the means for installing new software packages, upgrading existing software packages, updating the package list index, and even upgrading the entire Ubuntu system.

Some examples of popular uses for the apt utility include:

* **Install a Package**  
  Installation of packages using apt is quite simple. For example, to install the nmap network scanner, type the following:
* sudo apt install nmap

**Tip**  
You can specify multiple packages to be installed or removed, by separating them with spaces.

* **Remove a Package**  
  Removal of a package (or packages) is also straightforward. To remove the package installed in the previous example, simply type:
* sudo apt remove nmap

Adding the --purge option to apt remove will remove the package configuration files as well. This may or may not be the desired effect, so use with caution.

**Note**:  
While apt is a command-line tool, it is intended to be used interactively, and not to be called from non-interactive scripts. The apt-get command should be used in scripts (perhaps with the --quiet flag). For basic commands the syntax of the two tools is identical.

* **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 update
* **Upgrade packages**  
  Installed packages on your computer may periodically have upgrades available from the package repositories (e.g., security updates). To upgrade your system, first, update your package index with sudo apt update, and then type:
* sudo apt upgrade

For details on how to upgrade to a new Ubuntu release

Actions of the apt command, such as installation and removal of packages, are logged in the /var/log/dpkg.log log file.

Aptitude

Launching Aptitude with no command-line options will give you a menu-driven, text-based frontend to the 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.

Aptitude is best suited for use in a non-graphical terminal environment to ensure proper functioning of the command keys. You can start the menu-driven interface of Aptitude as a normal user by typing the following command at a terminal prompt:

sudo aptitude

When Aptitude starts, you will see a menu bar at the top of the screen and two panes below the menu bar. The top pane contains package categories, such as *New Packages* and *Not Installed Packages*. The bottom pane contains information related to the packages and package categories.

Using Aptitude for package management is relatively straightforward, and the user interface makes common tasks simple to perform. The following are examples of common package management functions as performed in Aptitude:

* **Install Packages**  
  To install a package, locate it 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 it has been marked for installation. Now press g to be presented with a summary of package actions. Press g again, and the package will be downloaded and installed. When finished, press Enter to return to the menu.
* **Remove Packages**  
  To remove a package, locate it in the *Installed Packages* package category by using the keyboard arrow keys and the Enter key. Highlight the package you want 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 the package will be removed. When finished, press Enter to return to the menu.
* **Update Package Index**  
  To update the package index, simply press the u key.
* **Upgrade Packages**  
  To upgrade packages, first update the package index as detailed above, and then press the U key to mark all packages with updates. Now press g, which will present you with a summary of package actions. Press g again to begin the download and installation. When finished, press Enter to return to the menu.

The first column of information displayed in the package list (in the top pane) lists the current state of the package (when viewing packages). It uses the following key to describe the package state:

* **i**: Installed package
* **c**: Package not installed, but package configuration remains on the system
* **p**: Purged from system
* **v**: Virtual package
* **B**: Broken package
* **u**: Unpacked files, but package not yet configured
* **C**: Half-configured - configuration failed and requires fix
* **H**: Half-installed - removal failed and requires a fix

To exit Aptitude, simply press the q key and confirm you wish to exit. Many other functions are available from the Aptitude menu by pressing the F10 key.

Command Line Aptitude

You can also use Aptitude as a command-line tool, similar to apt. To install the nmap package with all necessary dependencies (as in the apt 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

Consult the Aptitude manpages for full details of Aptitude’s command-line options.

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. **APT and Aptitude are newer, and layer additional features on top of**dpkg**.** This section covers using dpkg to manage locally installed packages:

* To list *all* packages in the system’s package database, installed and uninstalled, from a terminal prompt type:
* dpkg -l
* Depending on the number of packages on your system, this can generate a large amount of output. Pipe the output through grep to see if a specific package is installed:
* dpkg -l | grep apache2

Replace apache2 with any package name, part of a package name, or a regular expression.

* To list the files installed by a package, in this case the ufw package, enter:
* dpkg -L ufw
* If you are unsure which package installed a file, dpkg -S may be able to tell you. For example:
* dpkg -S /etc/host.conf
* base-files: /etc/host.conf

The output shows that the /etc/host.conf belongs to the base-files package.

**Note**:  
Many files are automatically generated during the package install process, and even though they are on the filesystem, dpkg -S may not know which package they belong to.

* You can install a local .deb file by entering:
* sudo dpkg -i zip\_3.0-4\_amd64.deb

Change zip\_3.0-4\_amd64.deb to the actual file name of the local .deb file you wish to install.

* You can uninstall a package by:
* sudo dpkg -r zip

**Caution**:  
Uninstalling packages using dpkg, is *NOT* recommended in most cases. It is better to use a package manager that handles dependencies to ensure that the system is in a consistent state. For example, using dpkg -r zip will remove the zip package, but any packages that depend on it will still be installed and may no longer function correctly.

For more dpkg options see the manpage: man dpkg.

APT configuration

Configuration of the APT system repositories is stored in the /etc/apt/sources.list file and the /etc/apt/sources.list.d directory. An example of this file is referenced here, along with information on adding or removing repository references from the file.

You can edit the file to enable and disable repositories. For example, to disable the requirement of inserting the Ubuntu CD-ROM whenever package operations occur, simply comment out the appropriate line for the CD-ROM, which appears at the top of the file:

# no more prompting for CD-ROM please

# deb cdrom:[DISTRO-APT-CD-NAME - Release i386 (20111013.1)]/ DISTRO-SHORT-CODENAME main restricted

Extra repositories

In addition to the officially-supported package repositories available for Ubuntu, there are also community-maintained repositories which add thousands more packages for potential installation. Two of the most popular are the *universe* and *multiverse* repositories. These repositories are not officially supported by Ubuntu, but because they are maintained by the community they generally provide packages which are safe for use with your Ubuntu computer.

**Note**:  
Packages in the *multiverse* repository often have licensing issues that prevent them from being distributed with a free operating system, and they may be illegal in your locality.

**Warning**:  
Be advised that neither *universe* nor *multiverse* contain officially-supported packages. In particular, there may not be security updates for these packages.

Many other package sources are available – sometimes even offering only one package, as in the case of packages provided by the developer of a single application. You should always be very careful and cautious when using non-standard package sources/repos, however. Research the packages and their origins carefully before performing any installation, as some packages could render your system unstable or non-functional in some respects.

By default, the *universe* and *multiverse* repositories are enabled. If you would like to disable them, edit /etc/apt/sources.list and comment out the following lines:

deb <http://archive.ubuntu.com/ubuntu> DISTRO-SHORT-CODENAME universe multiverse

deb-src <http://archive.ubuntu.com/ubuntu> DISTRO-SHORT-CODENAME universe multiverse

deb <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME universe

deb-src <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME universe

deb <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME-updates universe

deb-src <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME-updates universe

deb <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME multiverse

deb-src <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME multiverse

deb <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME-updates multiverse

deb-src <http://us.archive.ubuntu.com/ubuntu/> DISTRO-SHORT-CODENAME-updates multiverse

deb <http://security.ubuntu.com/ubuntu> DISTRO-SHORT-CODENAME-security universe

deb-src <http://security.ubuntu.com/ubuntu> DISTRO-SHORT-CODENAME-security universe

deb <http://security.ubuntu.com/ubuntu> DISTRO-SHORT-CODENAME-security multiverse

deb-src <http://security.ubuntu.com/ubuntu> DISTRO-SHORT-CODENAME-security multiverse

Automatic updates

The unattended-upgrades package can be used to automatically install updated packages and can be configured to update all packages or just install security updates. First, install the package by entering the following in a terminal:

sudo apt install unattended-upgrades

To configure unattended-upgrades, edit /etc/apt/apt.conf.d/50unattended-upgrades and adjust the following to fit your needs:

Unattended-Upgrade::Allowed-Origins {

"${distro\_id}:${distro\_codename}";

"${distro\_id}:${distro\_codename}-security";

// "${distro\_id}:${distro\_codename}-updates";

// "${distro\_id}:${distro\_codename}-proposed";

// "${distro\_id}:${distro\_codename}-backports";

};

Certain packages can also be excluded and therefore will not be automatically updated. To block a package, add it to the list:

Unattended-Upgrade::Package-Blacklist {

// "vim";

// "libc6";

// "libc6-dev";

// "libc6-i686";

};