How To Manage Packages In Ubuntu and Debian With Apt-Get & Apt-Cache

Authored by: ***ASPHostServer Administrator*** *[asphostserver@gmail.com]*
Saved From: *http://faq.asphosthelpdesk.com/article.php?id=201*

What is Apt-Get?

Apt is a command line frontend for the dpkg packaging system and is the preferred way of managing software from the command line for many distributions. It is the main package management system in Debian and Debian-based Linux distributions like Ubuntu.

While a tool called "dpkg" forms the underlying packaging layer, apt-get and apt-cache provide user-friendly interfaces and implement dependency handling. This allows users to efficiently manage large amounts of software easily.

In this guide, we will discuss the basic usage of apt-get and apt-cache and how they can manage your software. We will be practicing on an Ubuntu 12.04 cloud server, but the same steps and techniques should apply on any Debian-based distribution.

How To Update the Package Database with Apt-Get

Apt-get operates on a database of known, available software. It performs installations, package searches, and many other operations by referencing this database.

Due to this fact, before beginning any packaging operations with apt-get, we need to ensure that our local copy of the database is up-to-date.

Update the database with the following command. Apt-get requires administrative privileges for most operations:

```
sudo apt-get update
```

You will see a list of servers we are retrieving information from. After this, your database should be up-to-date.

How to Upgrade Installed Packages with Apt-Get

We can upgrade the packages on our system by issuing the following command:

```
sudo apt-get upgrade
```

For a more complete upgrade, you can use the "dist-upgrade" argument, which attempts intelligent dependency resolution for new packages and will upgrade essential programs at the expense of less important ones:

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

How to Install New Packages with Apt-Get

If you know the name of the package you wish to install, you can install it by using this syntax:

```
sudo apt-get install package1 package2 package3 ...
```

You can see that it is possible to install multiple packages at one time, which is useful for acquiring all of the necessary software for a project in one step.

It is important to understand that apt-get not only installs the requested software, but also any software needed to install or run it.

We can install the full "vim" text editor package by typing:

```
sudo apt-get install vim
```

How to Delete a Package with Apt-Get

To remove a package from your system, you can issue the following command:

```
sudo apt-get remove package_name
```

This command removes the package, but keeps the configuration files in case you install the package again later. This way, your settings will remain intact, even though the program is not installed.

If this is not the desired outcome, and you would like to clean out the configuration files as well as the program, use the following syntax:

```
sudo apt-get purge package_name
```

This uninstalls the package and removes any configuration files associated with the package.

To remove any packages that were installed automatically to support another program, that are no longer needed, type the following command:

```
sudo apt-get autoremove
```

You can also specify a package name after the "autoremove" command to uninstall a package and its dependencies.

Common Apt-Get Option Flags

There are a number of different options that can be specified by the use of flags. We will go over some common ones.

To do a "dry run" of a procedure in order to get an idea of what an action will do, you can pass the "-s" flag for "simulate":

```
sudo apt-get install -s htop

Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  strace ltrace
The following NEW packages will be installed:
  htop
0 upgraded, 1 newly installed, 0 to remove and 118 not upgraded.
Inst htop (1.0.1-1 Ubuntu:12.04/precise [amd64])
Conf htop (1.0.1-1 Ubuntu:12.04/precise [amd64])
```

In place of actual actions, you can see a "Inst" and "Conf" section specifying that there is where the package would be installed and configured if the "-s" was removed.

If you do not want to be prompted to confirm your choices, you can also pass the "-y" flag to automatically assume "yes" to questions.

```
sudo apt-get remove -y htop
```

If you would like to download a package, but not install it, you can issue the following command:

```
sudo apt-get install -d packagename
```

The files will be located in "/var/cache/apt/archives".

If you would like to suppress output, you can pass the "-qq" flag to the command:

```
sudo apt-get remove -qq packagename
```

How to Find a Package with Apt-Cache

The apt packaging tool is actually a suite of related, complimentary tools that are used to manage your system software.

While "apt-get" is used to upgrade, install, and remove packages, "apt-cache" is used to query the package database for package information.

You can use the following command to search for a package that suits your needs. Note that apt-cache doesn't usually require administrative privileges:

```
apt-cache search what_you_are_looking_for
```

For instance, if we wanted to find "htop", an improved version of the "top" system monitor, we can type the following:

```
apt-cache search htop

aha - ANSI color to HTML converter
```

```
htop - interactive processes viewer
```

We can search for more generic terms also. In this example, we'll look for mp3 conversion software:

```
apt-cache search mp3 convert

abcde - A Better CD Encoder
cue2toc - converts CUE files to cdrdao's TOC format
dir2ogg - audio file converter into ogg-vorbis format
easytag - viewing, editing and writing ID3 tags
hpodder - Tool to scan and download podcasts (podcatcher)
id3v2 - A command line id3v2 tag editor
kid3 - KDE MP3 ID3 tag editor
kid3-qt - Audio tag editor
. . .
```

How to View Package Information with Apt-Cache

To view information about a package, including an extended description, use the following syntax:

```
apt-cache show package_name
```

This will also provide the size of the download and the dependencies needed for the package.

To see if a package is installed and to check which repository it belongs to, we can issue:

```
apt-cache policy package_name
```

Conclusion

You should now know enough about apt-get and apt-cache to manage most of the software on your server.

While it is sometimes necessary to go beyond these tools and the software available in the repositories, most software operations can be managed by these tools.

If you are planning on spending time in a Debian-based environment, it is essential to have a working knowledge of these tools.