6.3.2 Advanced Packaging Tool (apt-get)

Click one of the buttons to take you to that part of the video.

Advanced Packaging Tool (apt-get) 0:00-0:13

The Advanced Packaging Tool Suite of tools includes the apt-get utility, which is really the equivalent of the yum utility on an RPM system. I love apt-get.

---

apt-get 0:14-3:33

It's so handy, because it automatically downloads and installs packages for you. It also automatically calculates all of the dependencies for the package you want to install and then automatically downloads and installs those dependent packages for you as well. It's much better than trying to do it alone with a dpkg command.

You need to be aware of this file right here: /etc/app/sources.list. This file is important because it defines which repositories the apt-get utility can get packages from to install them. And just as with yum, these repositories can reside on a local optical disc, such as your installation DVD.

That repository could be on your local hard disc drive, or more than likely, the way I always do it, is that that repository is on a server on the internet typically maintained by your Linux distribution's vendor.

In this example of the sources.list file, you can see a list of all the different repositories where apt-get can get package files from to install on your system.

I want to point out to you that within this particular file, there are two different types of repositories. Your standard package repositories, where apt-get can get actual Debian packages, always begin with a prefix of deb.

On the other hand, if you wanted to, you could have the apt-get utility not download an actual package file, but the source code for the package file. It gets those from a different type of repository. These are called source file repositories. They're identified in the sources.list file by the prefix deb-src, as you can see right here.

The syntax for using apt-get is pretty straight-forward. We run apt-get, followed by a list of options, as well as a command, and then the name of the package that we want to perform those actions on.

For example, if you wanted to install a new package with apt-get, you would simply type apt-get and then the action, the command, install, and then the name of the package that you want to install.

In the example that you see here, we first use sudo to elevate privileges and then we run the apt-get install nmap command. When I do this, the apt-get utility will go out to one of the repositories that are configured in the sources.list file that we just looked at.

It will download it. It will then query that package for dependencies and if there are dependencies that have not been installed on the system, then apt-get will go back to the repository and download those as well and install them on the system and then install the actual package itself.

In this case, we're going to install the nmap utility. Just as with yum, there are many more tasks that you can complete with apt-get besides just installing software.

For example, if you want to upgrade a package, use the apt-get upgrade command. If you want to uninstall a package, you use the apt-get remove command. And if you want to just upgrade all of the packages on your system with the latest upgrades available, you enter apt-get dist-upgrade.

All of the APT tools that we've looked at thus far have been command line utilities. However, the APT suite also provides a text-based menu driven utility that you can use to manage Debian packages on your system, and it's kind of handy. It's called Aptitude.

Aptitude 3:34-5:24

Be aware that before you can actually use Aptitude, you're probably going to first need to install it on the system. Most of the Debian distributions that I've worked with do not include Aptitude by default in the base installation.

What you're going to have to do in order to use Aptitude is first run this command. Use sudo to elevate privileges then run apt-get install and specify the aptitude package.

That will go out and download Aptitude from your repository and install it. Once that's done, you can run Aptitude by simply running the aptitude command at the shell prompt, and when you do, the interface that you see here is displayed.

Within this interface you can use aptitude to do just about anything you can do with dpkg or apt-get. You can install packages, uninstall packages, you can update packages.

For example, if I wanted to install a new package, I would arrow down to this option right here, Not Installed Packages, and then I'd browse through the list to find the package that I want to install, and then I would press g to install it.

Be aware that you don't have to use the menu driven interface with Aptitude that we just saw if you don't want to. Instead, you can use aptitude as a command line utility, much like apt-get. In fact, it will do a lot of the same things that apt-get will do.

First of all, if we want to install a package name, we enter aptitude install followed by the package name. We can use aptitude remove to uninstall a package.

We can use aptitude purge to first uninstall the package, and then also delete all of its configuration and data files. In other words, we're purging that package completely off the system.

We can use the aptitude update command to update the list of available packages from the repository. We can use the aptitude full-upgrade command to upgrade all of the installed packages on your system to the latest versions, and we can also use aptitude search to search for packages that contain the specified search term here.

Summary 5:25-5:32

That's it for this lesson. In this lesson, we reviewed APT tools that you can use to manage Debian packages. We first looked at apt-get, and then we looked at the aptitude command.
