Chapter 10 APT - Notes

10.2 Introduction

For use on Debian-based systems, **APT** (**A**dvanced **P**ackaging **T**ool) set of programs provides higher level of intelligent services for using underlying **dpkg** program, plays same role as **yum** on Red Hat-based systems. Main utilities: **apt-get**, **apt-cache**. Can automatically resolve dependencies when installing, updating, removing packages. Accesses external software repositories, synchronizing with them, retrieving/installing software as needed.

10.3 Learning Objectives:

- Explain what APT is.
- Perform package queries.
- Clean up system information about packages.
- Install, remove, and upgrade packages using **apt-get**.

10.4 What is APT?

APT not a program in itself: stands for **A**dvanced **P**ackaging **T**ool, which includes number of utilities, eg. **apt-get**, **apt-cache**. These, of course, in turn, invoke lower-level **dpkg** program.

APT system works with Debian packages whose files have `.deb` extension. Many distributions that have descended from Debian (eg. Ubuntu, Linux Mint) which have adopted Debian packaging system with no essential modification. Not uncommon to use repository on more than one Debian-based Linux distribution.

Going to ignore graphical interfaces (on computer), eg. Synaptic, Ubuntu Software Center, or other older frontends to APT, eg. aptitude.

However, excellent Internet-based resources can be found on Debian packages webpage and Ubuntu packages webpage. These databases let you search for packages, examine their contents, and download.

10.5 apt-get

apt-get: main APT command line tool for package management. Can be used to install, manage, upgrade individual packages, or the entire system. Can even upgrade distribution to completely new release, which can be difficult task.

There are even (imperfect) extensions that let **apt-get** work with **rpm** files.

Like **yum** and **zypper**, works with multiple remote repositories.

10.6 Queries

- Search repository for package named `apache2`:

```
$ apt-cache search apache2
```

- Display basic information about `apache2` package:

```
$ apt-cache show apache2
```

- Display more detailed information about `apache2` package:

```
$ apt-cache showpkg apache2
```

- List all dependent packages for `apache2`:

```
$ apt-cache depends apache2
```

- Search repository for file named `apache2.conf`:

```
$ apt-file search apache2.conf
```

- List all files in `apache2` package:

```
$ apt-file list apache2
```

10.7 Cleaning Up

To get rid of any packages that are not needed anymore, such as older Linux kernel versions:

```
$ sudo apt-get autoremove
```

To clean out cache files and any archived package files that have been installed:

```
$ sudo apt-get clean
```

Over time, may be some packages that are no longer needed at all. Can use `autoremove` to take them off system.

Furthermore, can reduce amount of data stored in `/var/cache/apt*` without causing any harm by using `clean` as any `apt-get [update|upgrade]` command will bring in most current information back into system.

10.8 Installing/Removing/Upgrading

apt-get program work horse of installing, removing, upgrading packages:

- Synchronize package index files with their repository sources. Indexes of available packages fetched from location(s) specified in `/etc/apt/sources.list`:

```
$ sudo apt-get update
```

- Install new packages or update an already installed package:

```
$ sudo apt-get install [package]
```

- Remove package from system without removing its configuration files:

```
$ sudo apt-get remove [package]
```

- Remove package from system, as well as its configuration files:

```
$ sudo apt-get --purge remove [package]
```

- Apply all available updates to packages already installed:

```
$ sudo apt-get upgrade
```

- Do a **smart upgrade** that will do a more thorough dependency resolution and remove some obsolete packages and install new dependencies:

```
$ sudo apt-get dist-ugprade
```

This will not update to whole new version of Linus distribution, as is commonly misunderstood.

- Note: must **update** before **upgrade**, unlike with **yum**, where `update` argument does both steps (update repositories and then upgrades packages). Can be confusing to habitual **yum** users on Debian-based systems.

- Get rid of any packages not needed anymore, such as older Linux kernel versions:

```
$ sudo apt-get autoremove
```

- Clean out cache files and any archived package files that have been installed:

```
$ sudo apt-get clean
```

This can save lot of space.

##

Back to top

Previous Chapter - Table of Contents - Next Chapter