Managing Packages with APT

A Presentation by David Lloyd

About the Author

- User of Debian GNU/Linux for approximately a year
- User of Linux for at least 8 years
- Currently employed and working on a distribution based on Debian GNU/Linux

About Apt - 1

- **A**dvanced **P**ackaging **T**ool
  - Originally built as a tool for Debian GNU/Linux
  - Conectiva ported a subset of the APT tools for RPM based systems

About APT - 2

- Manages binary packages
  - A package is a program or group of programs to provide particular functionality
  - A binary package is a package that contains pre-compiled or pre-built code

About APT - 3

- Why use it?
  - Pros:
    - Eases system administration
    - Eases dependency problems
  - Cons:
    - Removes a measure of control from the system administrator
    - Integrating APT with non-APT sources can be dangerous (but not impossible)

About APT - 4

- APT is **not**:
  - A package format (such as an RPM, DEB package or FreeBSD “.tar.gz” package)
  - A total replacement for source built packages
  - A tool to manage system configuration or the pre/post configuration of the underlying package tools

APT Components - 1

- The base set of APT tools:
  - apt-get
  - apt-cache
- With corresponding config files:
  - apt.conf
  - sources.list
- And some administrative directories
  - The archive and lists directories

APT Components – 2

- Advanced tools:
  - apt-move
  - apt-cache
  - auto-apt
  - apt-spy
- GUI Tools
  - synaptic, kpackage and others
- Some of these tools are Debian specific

APT Components – 3

- The APT Repository
  - A repository of packages
  - A list of available packages
  - A description of package dependencies

Installing APT - Debian

- Debian installs all the following by default:
  - apt-get
  - apt-cache
  - apt-config
  - apt-cdrom
- APT is an **essential** package

Installing APT for RPM

- Visit:
  - http://apt4rpm.sourceforge.net

Note:

You will need to install APT not using APT unless you happen to be using Conectiva Linux. This is a **totally** and **utterly** frustrating experience and should be enough to convince you to use APT anyway.

Configuring APT

- A basic configuration can be achieved simply by setting up the **sources.list**

Configuring APT - Debian

- Use:
  - apt-config

Configuring APT – Other

- Conectiva
  - Configured by default
- RedHat
  - Follow the instructions from the apt4rpm site
- SuSE
  - Advisable to setup APT on a newly installed system
- There is a **linuxconf** module for setup

Using APT

- Basic steps:
  1. Synchronise the **apt** database
  2. Update the existing packages
  3. Install or remove packages as you wish
  4. Repeat step 1 at regular intervals
- You may skip the first step at your discretion

Using APT – Satisfying Dependencies

- Satisfying dependencies
  - Sometimes **apt-get** will stop saying that it wants to install a particular package but was not instructed to
  - This happens when a **dependency** must be satisfied however satisfying it may break your system or download an unreasonable amount
  - Use **apt-get -f install**

Installing – Advanced

- You can specify versions:
  - **apt-get install gnome=1.2**
- You can override what APT attempts to do:
  - **apt-get install +gnome -kdebase**
- You can use regular expressions:
  - **apt-get install gnome* ?term***

Differences?

Debian

- Availability of advanced tools
- The underlying packaging system has different features to the RPM format
- Dependencies are made against **packages** not **files**
- A single, definitive source for Debian packages

RPM

- Advanced tools are not yet available for RPM
- Underlying package system dependencies have a fundamental flaw
- No definitive source for RPM's – there are just too many distributions

Synaptic – A GUI Tool

- A good GUI is:
  - **synaptic**
- To install:
  1. Install APT
  2. apt-get install synaptic
- **Synaptic** is able to empower all desktop users without involving them in a dependency quagmire

Searching for Available Packages

- Use **apt-cache search**
  - Under Debian multiple keywords can be used
  - Under other systems only a single keyword can be used
- Use **apt-cache show**
  - Shows the description of the packages

Finding out Dependency Information

- Use **apt-cache depends**
  - Shows what a package depends on
- Use **apt-cache rdepends**
  - Shows what depends on the package

Performing Updates

- Use **apt-get upgrade**
  - All installed packages will be upgraded
  - Upgrades to highest priority possible or latest package
  - Will not remove or delete any packages
  - Will not install any new packages
- You most likely want to run an **apt-get update** before upgrading

Performing Upgrades

- Use **apt-get dist-upgrade**
  - Updates all available packages
  - Removes and replaces packages if required
  - Installs new packages if required
- A **dist-upgrade** may cause difficulties in your system

Upgrade Differences

Debian

- Has three primary streams:
  - Stable
  - Testing
  - Unstable
- Has the concept of priority
- Has a single, strictly managed repository

RPM

- No concept of primary streams
  - You have to trust the repository maintainer not to have broken anything
- Has no concept of priority/half-instsalled or configured
  - RPMs are either installed or not
- No single, strictly managed repository

Stuff to Do at Home

- Investigate **apt-move**
  - A utility to create and manage a mirror of Debian packages
- Investigate **apt-proxy**
  - A tool used to create an APT proxy repository
- Use “equivalent packages” for source built installs

Conclusion

The Advanced Package Management Tool is a useful addition to any system administrator's arsenal.

It eases administration by resolving dependency problems with a standardised, well-recognised tool.

Although APT does replace the RPM format or DEB format, it is a useful “meta-package” tool that overcomes both these formats' deficiencies.