

# Package Management System

Presented By:

Amir Reza Niakan Lahiji

# What is package management system?

- A **package management** system is a collection of tools to automate the process of installing, upgrading, configuring, and removing software packages from a computer.
- Packages are distributions of software and metadata such as the software's full name, description of its purpose, version number, vendor, checksum, and a list of dependencies necessary for the software to run properly.
- Upon installation, metadata is stored in a local *package database*.

# Package Management System vs. Installer

Package Management System	Installer
Typically part of the operating system.	Each product comes bundled with its own installer.
Uses a single installation database.	Performs its own installation, sometimes recording information about that installation in a registry.
Can verify and manage all packages on the system.	Only works with its bundled product.
Single package management system vendor.	Multiple installer vendors.
Single package format.	Multiple installation formats.

# Package Management System Functions

- **Typical functions of a package management system include:**
  - Verifying file checksums to ensure correct and complete packages.
  - Verifying digital signatures to authenticate the origin of packages.
  - Applying file archives to manage encapsulated files.
  - Upgrading software with latest versions, typically from a software repository.
  - Grouping of packages by function to help eliminate user confusion.
  - Managing dependencies to ensure a package is installed with all packages it requires.

# Package Formats

- Each package manager relies on the format and metadata of the packages it can manage.
- Often a suit of tools manages the basic installation from these packages and other managers relies on them (Provide new functionalities).
  - `yum` relies on `rpm` as a backend
  - Synaptic Package Manager uses apt library
  - `apt` relies on `dpkg` as a backend

# Package Managers past and present

- Traditional Linux package management systems such as RPM present several problems for users.
  - Dependency problems
  - Finding packages
- Present Linux package management systems such as APT solve the problems
  - Internet-based repository

# Examples of package managers

- **RPM Package Manager**
  - The file format *RPM* is the baseline package format of the Linux Standard Base.
- **Advanced Packaging Tool (APT)**
  - APT was originally designed as a front-end for dpkg to work with Debian's .deb packages but it has since been modified to also work with the RPM Package Manager system via apt-rpm.
- **Synaptic**
  - is a GTK+ graphical user interface front-end to the Advanced Packaging Tool

# Working with APT package manager

- To search for a package
  - apt-cache search messenger
- Then to see information about that package:
  - apt-cache show pidgin
  - apt-cache depends pidgin
  - apt-cache showpkg pidgin
- To download source a package:
  - apt-get source pidgin
  - apt-get -b source pidgin

# Working with APT package manager

- To install a package
  - `apt-get install pidgin`
- To reinstall a package
  - `apt-get --reinstall install pidgin`
- To uninstall a package
  - `apt-get remove pidgin`
  - `apt-get --purge remove pidgin`
- To update package list
  - `apt-get update`

# Working with APT package manager

- To upgrade your all packages in your distro
  - `apt-get -u upgrade`
  - Cache Limit?
    - `vi /etc/apt/apt.conf.d/70debconf`
    - `APT::Cache-Limit "10000000";`
- To upgrade your distro
  - `apt-get -u dist-upgrade`
- To Remove unused package files
  - `apt-get clean`
    - `/var/cache/apt/archives/`
    - `/var/cache/apt/archives/partial/`
  - `apt-get autoclean`

# Working with APT package manager

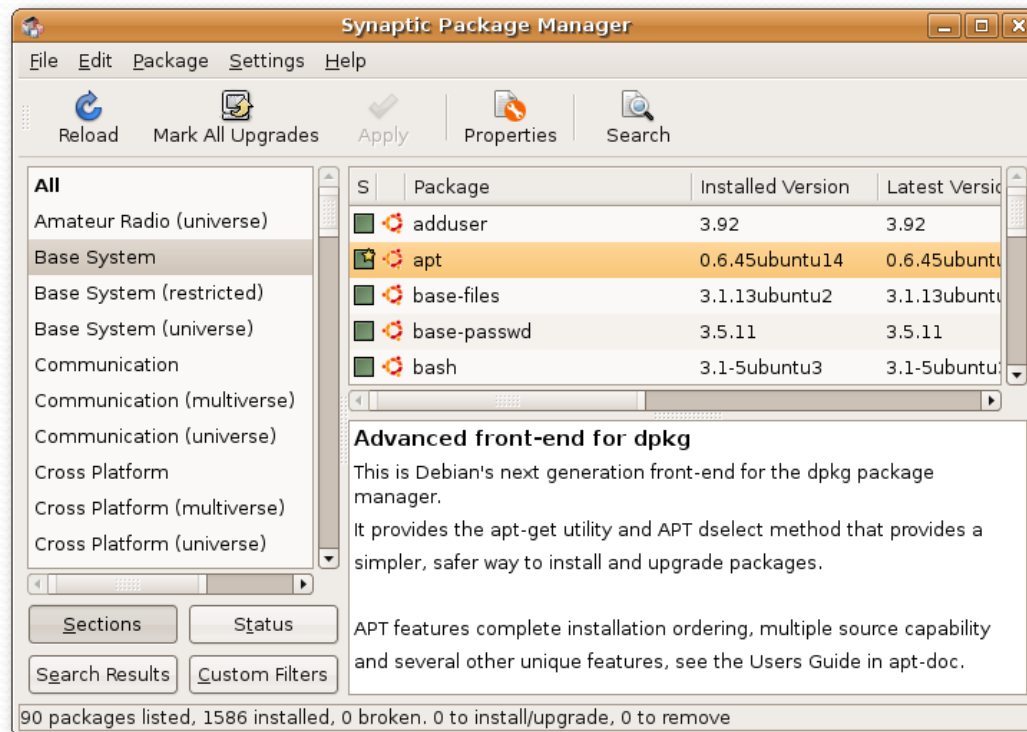
- apt repositories are specified in:
  - /etc/apt/sources.list
    - deb http://host/debian distribution section1 section2 section3
    - deb-src http://host/debian distribution section1 section2 section3
  - After editing this file you must run:
    - apt-get update

# Working with APT package manager

- If an installation breaks in the middle of the process and you find that it's no longer possible to install or remove packages, try :
  - # apt-get -f install
  - # dpkg --configure -a

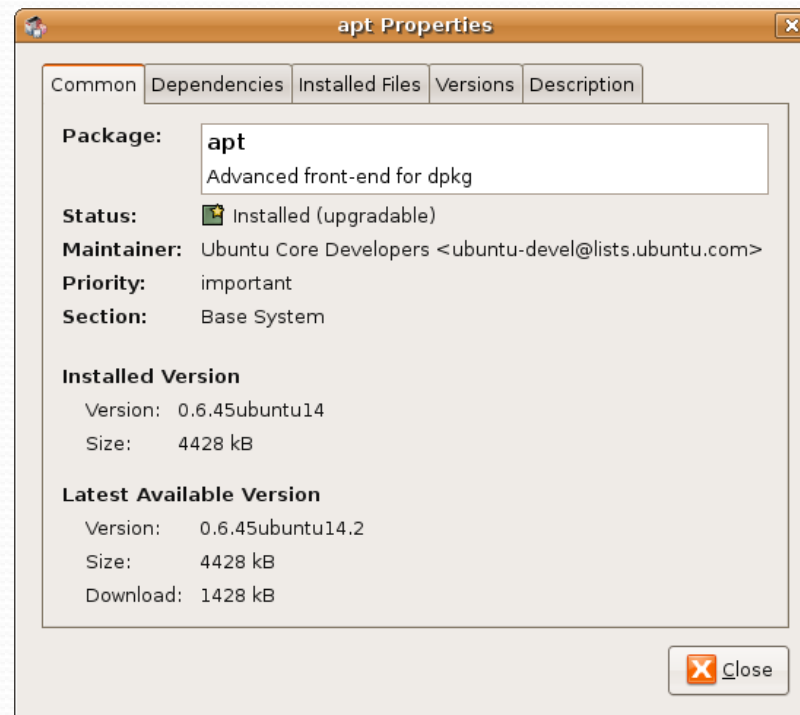
# Working with Synaptic

- To launch Synaptic, choose **System > Administration > Synaptic Package Manager**
















# Working with Synaptic

- To see more information about a package
  - Right Click on a **package** > **Properties**



# Working with Synaptic

-  Marked for installation
-  Marked for re-installation
-  Marked for upgrade
-  Marked for downgrade
-  Marked for removal
-  Marked for complete removal
- Not installed
-  Not installed (locked)
-  Installed
-  Installed (upgradable)
-  Installed (locked to the current version)
-  Broken
-  Not installed (new in repository)
-  Package is supported

# Working with Synaptic

- **To fix broken packages**
  - 'Broken packages' are packages that have unsatisfied dependencies
  - Choose **Edit > Fix Broken Packages** from the menu.
  - Choose **Apply Marked Changes** from the **Edit** menu or press **Ctrl + P**.
  - Confirm the summary of changes and click **Apply**.
- **To free disk space**
  - Settings -> Preferences -> Files
  - click **Delete Cached Package Files**