Package Managers

A **Package** is an archive of files required to install a particular application. Packages can include:

Features	Description
Binaries (executables)	Pre-compiled
Source code (ASCII text)	Not pre-compiled but ready to build.
Scripts	Run when a package is installed, updated, or removed; these may also be executed when other packages are installed, updated, or removed.
Digital signature	Verifies the integrity of the package being installed.
Configuration files	Required for the settings of an application.
Documentation files	Man, doc, info pages.
Information	Package metadata, such as the name, version, release, architecture, description of the package, dependencies, licensing, change logs, README files or other details.

Packages are often installed from a repository. A **repository** is a storage location from which software packages may be retrieved and installed on a computer. However, packages do not have to be installed from a repository.

The automation of installing, upgrading, configuring, and removing applications or packages is done using a **package manager.**

Types of Package Managers	Associated Commands
RPM Based Linux Distributions	`rpm` `yum`
Debian Based Linux Distributions	`dpkg` `apt-get`, `apt-cache`, `apt`
Solaris (Unix)	`pkgadd`, `pkgchk`, `pkginfo`, `pkgrm`

RPM Based Package Management

`rpm` (RPM Package Manager)

The `rpm` command is used to manage (**.rpm**) package files and is most commonly found on RPM based Linux variants such as Red Hat, Fedora, CentOS, openSUSE, or Scientific Linux. RPM was developed by Red Hat and was originally an acronym for Red Hat Package Manager. It provides a standard way for application developers to package software for distribution. The `rpm` command is used to **build, install, query, verify, update, and erase** individual software packages.

RPM package files are typically named accordingly:

- `name-version-release.architecture.rpm`
- `httpd-tools-2.4.6-7.el7.x86_64.rpm`

`yum` (Yellowdog Updater Modified)

The `yum` command is most commonly found on rpm based Linux distributions.

`yum` is essentially a wrapper for the `rpm` command which provides additional functionality in the form of **logging**, **history**, **installation from repositories** of RPM packages, system **updates**, **package removal**, and automatic package **dependency resolution**. The yum log is located at `/var/log/yum.log`.

The `yum` command can search numerous repositories (often called **repos**) for packages and their dependencies, then subsequently install them together to resolve dependency issues. Repositories are configured in the `/etc/yum.repos.d` directory in `.repo` files. These files are used to configure a repository location which can be used to install packages. Repos can be configured from many different locations such as local directories, network shares, ftp servers, web servers, etc.

Please note that Red Hat is currently testing the `dnf` *utility on Fedora distributions as a replacement for* `yum`.

Debian Package Management

dpkg (Debian Package)

`dpkg` is to Debian-based systems as `rpm` is to RPM based systems. It is used to **install**, **build**, **remove**, and **manage** Debian packages (**.deb**). Just as with `rpm`, `dpkg` does not natively support installing a package from a software repository. Rather, it is concerned with managing packages on the local system and, like `rpm`, relies on other utilities, such as `apt-get`, to retrieve packages from a repository.

The default log file for `dpkg` is `/var/log/dpkg.log`; however, this can be changed in `/etc/dpkg/dpkg.cfg`.

APT (Advanced Package Tool)

APT is a software suite and user interface which provides functionality similar to `yum` on RPM based systems. The APT utilities are used as wrappers around `dpkg` to determine available software packages in repositories and install desired packages. APT (the tool suite) should not be confused with `apt`, a specific utility within the APT suite which is discussed below. The majority of utilities provided in the APT suite are typically prefixed with "`apt-`".

apt-get

`apt-get` is a utility used to handle packages in software repositories.

- retrieving an updated list of available packages
- upgrading installed packages to newer versions
- installing software

apt-cache

In order to minimize the time and resources required, APT maintains a local cache which contains metadata describing which packages are available and where they are available (i.e. which repository contains the package). `apt-cache` allows the user to perform a number of operations against this cache. One common use of `apt-cache` is to search through available packages listed in the local cache based on keywords.

apt

`apt` is a newer, higher level utility for managing packages on Debian based systems. It does not offer all the functions provided by older more traditional utilities (e.g. `apt-get`, `apt-cache`), but it provides the most commonly used functions of the various `apt-*` tools in one tool. It is now provided by default on many Debian based

distributions, but its use is optional. A user may use the lower level tool options like `apt-get` interchangeably with `apt`.

`apt` logs installations, removals, and upgrades to the same files which `apt-get` and other lower level utilities do (e.g. `/var/log/apt/*`).

Please note that there are numerous other front end utilities (both text-based and graphical) which allow an administrator to manage packages using the APT suite.

Solaris 10 Package Management

The Solaris Operating System provides a set of utilities that interpret System V Release 4 (SVR4) formatted packages, which are provided by Sun and its vendors. Packages can be installed from a mounted CD, a remote package server, an administration file, an HTTP URL, or a spool directory (`/var/spool/pkg` by default). Packages can be managed via a graphical interface or the following command line tools:

`pkginfo`

Displays package software information. Without options, pkginfo displays information about packages that are installed.

`pkgchk`

Checks the installation of a software package.

`pkgadd`

Installs digitally signed or unsigned software packages. Using signed packages provides an additional layer of security since you can verify the digital signature prior to installing the software.

`pkgrm`

Removes software package(s). `pkgrm` deletes all files associated with the specified package(s), unless those files are used by other installed packages.

Please note that if a specific package is not provided, the `pkgrm` *command removes **all** packages.*

`pkgadd` and **`pkgrm`** commands do not generate logs in a standard logging location. These commands update the Solaris Product Registry database to maintain a record of software installed on the system. `pkgadd` also logs software installation in `/var/sadm/install/logs/`.

Command Examples

Task	RPM	Debian	Solaris
Given a file, identify the package to which it belongs	`rpm -qf <full path to file>`	`dpkg -S <full path to file>`	`pkgchk -l -p <file>`
Given a package name, list the files belonging to that package	`rpm -ql <package-name>`	`dpkg -L <package-name>`	`pkgchk -l <package-name> | grep Path`
Verify integrity of ALL packages	`rpm -Va`	`dpkg -V debsums`	`pkgchk` [1]
Verify integrity of one package	`rpm -V <package-name>`	`dpkg -V <package-name>`	`pkgchk <package-name>` [1]

[1] If no errors are found, the system prompt is returned. Otherwise, the `pkgchk` command reports the error.

External Resources

Resource	URL
Solaris package managers	https://web.archive.org/web/20171003213916/https://docs.oracle.com/cd/E26505_01/html/E29492/ewbej.html
Debian package managers	https://web.archive.org/web/20171003214029/https://wiki.debian.org/DebianPackageManagement
RPM package managers	https://web.archive.org/web/20171003214118/https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/ch-yum.html
A comparison of RPM vs. Debian based package managers	https://web.archive.org/web/20170821202539/https://help.ubuntu.com/community/SwitchingToUbuntu/FromLinux/RedHatEnterpriseLinuxAndFedora
A comparison of package manager commands across many distributions	https://web.archive.org/web/20170821202709/https://wiki.archlinux.org/index.php/Pacman/Rosetta