Package Installation

MODULE 5.2


rpm

Red Hat Package Manager

- RPM (Red Hat Package Manager) is a default open source and a popular package management utility for Red Hat based systems like (RHEL, CentOS and Fedora).
- The tool allows system administrators and users to install, update, uninstall, query, verify and manage system software packages in Unix/Linux operating systems.
- The RPM formerly known as .rpm file, includes compiled software programs and libraries needed by the packages. This utility only works with packages that built on .rpm format.

Facts about RPM

- RPM is free and released under GPL (General Public License).
- RPM keeps the information of all the installed packages under /var/lib/rpm database.
- RPM is the only way to install packages under Linux systems, if you've installed packages using source code, then rpm won't manage it.
- RPM deals with .rpm files, which contains the actual information about the packages such as: what it is, from where it comes, dependencies info, version info etc.

Where to find RPM packages

Below is the list of rpm sites, where you can find and download all **RPM** packages.

- http://rpmfind.net
- http://www.redhat.com
- http://freshrpms.net/
- http://rpm.pbone.net/

*Please remember you must be **root** user when installing packages in Linux, with the root privileges you can manage rpm commands with their appropriate options.*

Five basic modes for RPM command

- **Install**
- **Remove**
- **Upgrade**
- **Verify**
- **Query**
- **Build software archives**

Contents of .rpm package

File Identifier

- Version

Package signature

- Verifies the integrity of the package

Header

- Basic description
- Name
- Version
- Checksums
- Software archive
- scripts

Basic tasks of RPM

- Installing new software
- Erasing or removing unneeded packages
- Upgrading installed s/w packages
- Querying information about s/w packages
- Verifying installation of s/w packages

How to Check an RPM Signature Package

- Check the PGP signature of packages before installing them on Linux systems and make sure its integrity and origin is **OK**.
- Use the following command with **–checksig** (**check signature**) option to check the signature of a package.

```
rpm --checksig pidgin-2.7.9-5.el6.2.i686.rpm
```

- **How to Install an RPM Package**
  - For installing an rpm software package, use the following command with **-i** option.
  - For example, to install an rpm package called **pidgin-2.7.9-5.el6.2.i686.rpm**.
  - **RPM command and options**

    -i : install a package

    -v : verbose for a nicer display

    -h: print hash marks as the package archive is unpacked.

```
rpm -ivh pidgin-2.7.9-5.el6.2.i686.rpm
```

How to check dependencies of RPM Package before Installing

- To do a dependency check before installing or upgrading a package.
- For example, use the following command to check the dependencies of **BitTorrent-5.2.2-1-Python2.4.noarch.rpm** package. It will display the list of dependencies of package.

```
rpm -qpR BitTorrent-5.2.2-1-Python2.4.noarch.rpm
```

RPM command and options

1.-q : Query a package

2.-p : List capabilities this package provides.

3.-R: List capabilities on which this package depen

rpm -q --provides libc

rpm -q --requires rpm

Some common RPM commands and their functions include:

- rpm -i package.rpm: Install an RPM package.
- rpm -e package_name: Remove an RPM package.
- rpm -q package_name: Query for information about an installed package.
- rpm -U package.rpm: Upgrade an RPM package.
- rpm -ql package_name: List the files included in an installed package.
- rpm -qa: List all installed RPM packages.

While RPM provides granular control over packages, it does not automatically handle package dependencies, which can be cumbersome in complex software environments. This is where YUM comes into play.

- YUM is a high-level package management tool built on top of RPM. It simplifies the process of managing software packages by automatically resolving and installing dependencies.
- YUM maintains a repository of packages and their metadata, which allows it to efficiently install, update, and remove software while ensuring that all required dependencies are satisfied.

Some common YUM commands and their functions include:

- **yum install package_name:** Install a package and its dependencies.
- **yum remove package_name:** Remove a package and any packages that depend on it.
- **yum update:** Update all installed packages to their latest versions.
- **yum search keyword:** Search for packages based on a keyword.
- **yum list installed:** List all installed packages.

- YUM is widely used on Red Hat-based systems because it simplifies package management and makes it easier to maintain system stability and security.
- It provides a user-friendly interface to the underlying RPM system and is capable of fetching packages from remote repositories, ensuring that the system is up-to-date and secure.
- In summary, RPM is a lower-level package management tool used to work with individual software packages in the RPM format, while YUM is a higher-level tool that simplifies package management by handling dependencies, enabling system administrators to manage software more efficiently on Red Hat-based Linux systems.