### Package Management in Linux

Package management is a method of installing and maintaining (which includes updating and probably removing as well) software on the system.

In the early days of Linux, programs were only distributed as source code, along with the required man pages, the necessary configuration files, and more. Nowadays, most Linux distributors use by default pre-built programs or sets of programs called packages, which are presented to users ready for installation on that distribution. However, one of the wonders of Linux is still the possibility to obtain source code of a program to be studied, improved, and compiled.

##### **How package management systems work**

If a certain package requires a certain resource such as a shared library, or another package, it is said to have a dependency. All modern package management systems provide some method of dependency resolution to ensure that when a package is installed, all of its dependencies are installed as well.

##### **Packaging Systems**

Almost all the software that is installed on a modern Linux system will be found on the Internet. It can either be provided by the distribution vendor through central repositories (which can contain several thousands of packages, each of which has been specifically built, tested, and maintained for the distribution) or be available in source code that can be downloaded and installed manually.

Because different distribution families use different packaging systems (Debian: \*.deb / CentOS: \*.rpm / openSUSE: \*.rpm built specially for openSUSE), a package intended for one distribution will not be compatible with another distribution.

##### **High and low-level package tools**

In order to perform the task of package management effectively, you need to be aware that you will have two types of available utilities: low-level tools (which handle in the backend the actual installation, upgrade, and removal of package files), and high-level tools (which are in charge of ensuring that the tasks of dependency resolution and metadata searching -”data about the data”- are performed).

|  |  |  |
| --- | --- | --- |
| DISTRIBUTION | LOW-LEVEL TOOL | HIGH-LEVEL TOOL |
| Debian and derivatives | dpkg | apt-get / aptitude |
| CentOS | rpm | yum |
| openSUSE | rpm | zypper |

Let us see the descrption of the low-level and high-level tools.

* dpkg is a low-level package manager for Debian-based systems. It can install, remove, provide information about and build \*.deb packages but it can’t automatically download and install their corresponding dependencies.
* apt-get is a high-level package manager for Debian and derivatives, and provides a simple way to retrieve and install packages, including dependency resolution, from multiple sources using the command line. Unlike dpkg, apt-get does not work directly with \*.deb files, but with the package proper name.
* aptitude is another high-level package manager for Debian-based systems, and can be used to perform management tasks (installing, upgrading, and removing packages, also handling dependency resolution automatically) in a fast and easy way. It provides the same functionality as apt-get and additional ones, such as offering access to several versions of a package.
* rpm is the package management system used by Linux Standard Base (LSB)-compliant distributions for low-level handling of packages. Just like dpkg, it can query, install, verify, upgrade, and remove packages, and is more frequently used by Fedora-based distributions, such as RHEL and CentOS.
* yum adds the functionality of automatic updates and package management with dependency management to RPM-based systems. As a high-level tool, like apt-get or aptitude, yum works with repositories.

YUM Commands for Package Management

**What is YUM?**

YUM (Yellowdog Updater Modified) is an open source command-line as well as graphical based package management tool for RPM (RedHat Package Manager) based Linux systems. It allows users and system administrator to easily install, update, remove or search software packages on a system. YUM uses numerous third-party repositories to install packages automatically by resolving their dependencies issues.

1. Install a Package with YUM

To install a package called [Firefox 14](https://www.tecmint.com/install-firefox-14-in-rhel-centos-fedora/), just run the below command it will automatically find and install all required dependencies for Firefox.

**# yum install firefox**

The above command will ask confirmation before installing any package on your system. If you want to install packages automatically without asking any confirmation, use option -y as shown in below example.

**# yum -y install firefox**

**2. Removing a Package with YUM**

To remove a package completely with their all dependencies, just run the following command as shown below.

**# yum remove firefox**

Same way the above command will ask confirmation before removing a package. To disable confirmation prompt just add option -y as shown in below.

**# yum -y remove firefox**

**3. Updating a Package using YUM**

Let’s say you have outdated version of [MySQL](https://www.tecmint.com/category/databases/mysql/) package and you want to update it to the latest stable version. Just run the following command it will automatically resolves all dependencies issues and install them.

**# yum update mysql**

**4. List a Package using YUM**

Use the list function to search for the specific package with name. For example to search for a package called openssh, use the command.

**# yum list openssh**

To make your search more accurate, define package name with their version, in case you know. For example to search for a specific version openssh-4.3p2 of the package, use the command.

**# yum list openssh-4.3p2**

**5. Search for a Package using YUM**

If you don’t remember the exact name of the package, then use search function to search all the available packages to match the name of the package you specified. For example, to search all the packages that matches the word .

**# yum search vsftpd**

**6. Get Information of a Package using YUM**

Say you would like to know information of a package before installing it. To get information of a package just issue the below command.

**# yum info firefox**

7. List all Available Packages using YUM

To list all the available packages in the Yum database, use the below command.

**# yum list | less**

8. List all Installed Packages using YUM

To list all the installed packages on a system, just issue below command, it will display all the installed packages.

**# yum list installed | less**

9. Yum Provides Function

Yum provides function is used to find which package a specific file belongs to. For example, if you would like to know the name of the package that has the /etc/httpd/conf/httpd.conf.

**# yum provides /etc/httpd/conf/httpd.conf**

10. Check for Available Updates using Yum

To find how many of installed packages on your system have updates available, to check use the following command.

**# yum check-update**

11. Update System using Yum

To keep your system up-to-date with all security and binary package updates, run the following command. It will install all latest patches and security updates to your system.

**# yum update**

12. List all available Group Packages

In Linux, number of packages are bundled to particular group. Instead of installing individual packages with yum, you can install particular group that will install all the related packages that belongs to the group. For example to list all the available groups, just issue following command.

**# yum grouplist**

**Installed Groups:**

Administration Tools

DNS Name Server

Dialup Networking Support

Editors

Engineering and Scientific

FTP Server

Graphics

Java Development

Legacy Network Server

**Available Groups:**

Authoring and Publishing

Base

Beagle

Cluster Storage

Clustering

Development Libraries

Development Tools

Eclipse

Educational Software

KDE (K Desktop Environment)

KDE Software Development

13. Install a Group Packages

To install a particular package group, we use option as groupinstall. Fore example, to install “MySQL Database“, just execute the below command.

**# yum groupinstall 'MySQL Database'**

Dependencies Resolved

=================================================================================================

Package Arch Version Repository Size

=================================================================================================

Updating:

unixODBC i386 2.2.11-10.el5 base 290 k

Installing for dependencies:

unixODBC-libs i386 2.2.11-10.el5 base 551 k

Transaction Summary

=================================================================================================

Install 1 Package(s)

Upgrade 1 Package(s)

Total size: 841 k

Is this ok [y/N]: y

Downloading Packages:

Running rpm\_check\_debug

Running Transaction Test

Finished Transaction Test

Transaction Test Succeeded

Running Transaction

Installing : unixODBC-libs 1/3

Updating : unixODBC 2/3

Cleanup : unixODBC 3/3

Dependency Installed:

unixODBC-libs.i386 0:2.2.11-10.el5

Updated:

unixODBC.i386 0:2.2.11-10.el5

Complete!

14. Update a Group Packages

To update any existing installed group packages, just run the following command as shown below.

**# yum groupupdate 'DNS Name Server'**

15. Remove a Group Packages

To delete or remove any existing installed group from the system, just use below command.

**# yum groupremove 'DNS Name Server'**

16. List Enabled Yum Repositories

To list all enabled Yum repositories in your system, use following option.

**# yum repolist**

repo id repo name status

base CentOS-5 - Base enabled: 2,725

epel Extra Packages for Enterprise Linux 5 - i386 enabled: 5,783

extras CentOS-5 - Extras enabled: 282

mod-pagespeed mod-pagespeed enabled: 1

rpmforge RHEL 5 - RPMforge.net - dag enabled: 11,290

updates CentOS-5 - Updates enabled: 743

repolist: 20,824

16. List all Enabled and Disabled Yum Repositories

The following command will display all enabled and disabled yum repositories on the system.

**# yum repolist all**

repo id repo name status

C5.0-base CentOS-5.0 - Base disabled

C5.0-centosplus CentOS-5.0 - Plus disabled

C5.0-extras CentOS-5.0 - Extras disabled

base CentOS-5 - Base enabled: 2,725

epel Extra Packages for Enterprise Linux 5 - i386 enabled: 5,783

extras CentOS-5 - Extras enabled: 282

repolist: 20,824

17. Install a Package from Specific Repository

To install a particular package from a specific enabled or disabled repository, you must use –enablerepo option in your yum command. For example to [Install PhpMyAdmin 3.5.2](https://www.tecmint.com/install-phpmyadmin-3-5-2-for-apache-or-nginx-on-rhelcentos-6-35-8-fedora-17-12/) package, just execute the command.

**# yum --enablerepo=epel install phpmyadmin**

18. Interactive Yum Shell

Yum utility provides a custom shell where you can execute multiple commands.

**# yum shell**

19. Clean Yum Cache

By default yum keeps all the repository enabled package data in /var/cache/yum/ with each sub-directory, to clean all cached files from enabled repository, you need to run the following command regularly to clean up all the cache and make sure that there is nothing unnecessary space is using. We don’t want to give the output of the below command, because we like to keep cached data as it is.

**# yum clean all**

20. View History of Yum

To view all the past transactions of yum command, just use the following command.

**# yum history**

Loaded plugins: fastestmirror

ID | Login user | Date and time | Action(s) | Altered

-------------------------------------------------------------------------------

10 | root | 2012-08-11 15:19 | Install | 3

9 | root | 2012-08-11 15:11 | Install | 1

8 | root | 2012-08-11 15:10 | Erase | 1 EE

7 | root | 2012-08-10 17:44 | Install | 1

6 | root | 2012-08-10 12:19 | Install | 2

5 | root | 2012-08-10 12:14 | Install | 3

4 | root | 2012-08-10 12:12 | I, U | 13 E<

3 | root | 2012-08-09 13:01 | Install | 1 >

2 | root | 2012-08-08 20:13 | I, U | 292 EE

1 | System | 2012-08-08 17:15 | Install | 560

history list