

# Introduction to Package Management

There are many Linux distribution system out there and one common way to differentiate between them is by using package managers .

Red Hat Package managers i.e. RPM is a base package managers for some Linux distribution systems like Centos , Fedora and Red hat enterprise Linux(RHEL). ubuntu distribution Linux system that is way more popular uses Debian package manager i.e. (.Deb) . there are a plenty of more Linux distribution system like Linux mint, ubuntu ,Debian ,pure OS etc. that uses Debian package managers.

## What is a package managers?

Software Package are nothing but a compressed archived file that contains everything that is needed to run a software.

*what it contains?*

1. Software Binaries
2. METADATA
3. Configuration Files

*It contains the metadata (i.e. all information about the software) , the configuration files that is needed to run a software and some of the software binaries.*

*Let's suppose we have to install a Gimp software ( GIMP is an acronym for GNU Image Manipulation Program \. It is a freely distributed program for such tasks as photo retouching, image composition and image authoring) , then we just have to install a software package for that. lets say we have a ubuntu system and the package is Gimp.deb package ,then just by installing this package do you think we will be able to use Gimp?\**

The answer is no ☹️

you will get some dependences errors while installing gimp .so why is this so ?? and how we can solve this ?

```
[~]$ dpkg -i gimp.deb
(Reading database ... 419857 files and directories
currently installed.)
Preparing to unpack gimp.deb ...
Unpacking gimp (2.10.8-2) over (2.10.8-2) ...
dpkg: dependency problems prevent configuration of
gimp:
  gimp depends on libgimp2.0 (>= 2.10.8); however:
    Version of libgimp2.0 on system is 2.8.22-1.
dpkg: error processing package gimp (--install):
  dependency problems - leaving unconfigured
Processing triggers for gnome-menus (3.13.3-
11ubuntu1.1) ...
Processing triggers for desktop-file-utils
(0.23+linuxmint6) ...
Processing triggers for mime-support (3.60ubuntu1)
...
Processing triggers for man-db (2.8.3-2ubuntu0.1)
...
Errors were encountered while processing:
  gimp
```

## Functions of Package Managers

1. Package Integrity and Authenticity (It gives a very secure way to install packages & software by Authenticity of different webs)
2. Simplified Package Management ( it gives a easy way to managing package via easy update, delete and install them )
3. Grouping Packages ( make a group of packages by using functions )

Manage Dependencies (helps us to not fall in dependencies hell by managing all dependencies )

## **Different types of package managers**

1. For a Debian based distribution system i.e. Linux mint , ubuntu , Debian ,pure OS etc. The base package manager is DPKG (Debian package). the frontend package that is generally used is APT its like cherry on the ice-cream top. There are some traditional frontend package managers that is also used with DPKG is APT-GET.
2. For a RPM based distribution system like fedora, centos ,RHEL , we use Red-hat package managers (rpm) . The base package manager for a RPM based distribution system is RPM. The frontend package managers that is used more often with rpm is YUM ( YELLOW DOG Updater, Modified) .The more advanced version that we uses more often as a frontend package manger is DNF.

## **Working With RPM & YUM**

Now that we are using Red-Hat package manager to manage distribution system like CentOS , fedora ,RHEL . The file extensions that we use for markup RPM Package is .RPM.

The basic modes of operation that we will use while working with RPM are-

1. Installation — `rpm -ivh telnet.rpm` (-i is used for installation , telnet is a package name , vh is used to print info of the command)
2. Uninstalling — `rpm -e telnet.rpm`
3. Upgrade — `rpm -Uvh telnet.rpm` (-U is used to upgrade package)
4. Query — `rpm -q telnet.rpm` ( *a database is there to manage all rpm package in the repo , to get information about a certain package of database we use this command*)
5. Verifying — `rpm -Vf <path to file>` ( it is used to check that a certain package that is installed in our local system is same or not with a main package , means does we have used a secure pathway to install our package ,do we have installed some other package)

After all these RPM features , rpm lacks in managing the dependencies of the software so for the rescue of it , YUM comes in the way .

## **YUM Package Manager**

1. YUM is a High Level Package Manager that uses RPM under the hood
2. YUM manages all the dependencies' problem and makes it Automatic Dependency Resolution.

3. A Software Repositories is made under the file directory (/etc/yum.repos.d) *which contains all the rpm extensions file along with dependencies list.*

4. YUM works with RPM Based Distros like centos etc.

How YUM works under the hood?

A software repositories is made under the file directory (/etc/yum.repos.d) *which contains all the rpm extensions file along with dependencies list.* Software repositories can be in our local system or it can be on remote where we access it by (HTTP/HTTPS/FTP). YUM uses Rpm under the hood to install some of their dependencies .

In some distribution systems like RHEL , we got a pre-installed software repositories at a location of (/etc/yum.repos.d/redhat.repo). *In some cases where some versions of the packages needs to be updated or completely changes then what should we do in that case . should we create our own custom software repo in distribution systems like RHEL .*

The answer is yes ☹️

yes, we can create our custom software repos. in this path I have created my own NGINX custom repo /etc/yum.repos.d/nginx.repo.

**Now Let's learn some YUM commands**

1. Installing — yum install -y httpd ( here we are trying to install httpd service by YUM)
2. yum repolist ( this command is used to list out all the repos)
3. yum provides scp ( this command is used to list out the package name for a specific command to run) scp is the argument in the command .
4. yum remove httpd — ( this command is used to remove a package)
5. yum update telnet — ( this command is used to update packages)
6. yum update — (this command is used to update all packages in the repo)

## **Working With DPKG & APT**

*Now that we are familiar with the rpm-based package managers let's check out the equivalent on Debian distros like pure os, Debian i.e. DPKG. The file extensions of a DPKG is .Deb*

DPKG is a low level package manager similar to RPM that uses APT and APT-GET under the hood to manages all the dependencies of a package.

The basic modes of operation that we will use while working with DPKG are-

1. Installation / Upgrade — `dpkg -i telnet.deb`
2. Uninstalling — `dpkg -r telnet.deb`
3. List — `dpkg -l telnet` ( *to list the version number of packages and a short description*)
4. Status — `dpkg -s telnet`( *to list out the status of a package which is installed*)
5. Verifying — `dpkg -p <path to file>`(\*it is used to check that a certain package that is installed in our local system is same or not with a main package , means does we have used a secure pathway to install our package ,do we have installed some other package)\*

As in Fig 1 example that has been given above, in that the installation of Gimp software has failed because we have used DPKG(a low level package manager )which is not capable of managing all the dependencies used for the Gimp software. So we will use APT for that or we can use APT-GET. but APT will be more good to use.

To install Gimp without any error use APT /APT-GET

*apt install gimp or apt-get install gimp*

How APT works under the hood?



A software repositories is made under the file directory (/etc/apt/sources.list) *which contains all the DPKG extensions file along with dependencies list*. Software repositories can be in our local system or it can be on remote where we access it by (HTTP/HTTPS/FTP). APT uses DPKG under the hood to install some of their dependencies .

## **Now Let's learn some APT commands**

1. apt install telnet ( to install a package )
2. apt remove telnet ( to remove a package)
3. apt search telnet ( to search for a package in the repo)
4. apt list | grep telnet ( to list out the package with a short description)
5. apt update ( to download package information or to refresh the repo. a good time to run this when we add new resources or immediately after adding the OS)
6. apt upgrade ( to upgrade all the packages in the repo)
7. apt edit-sources ( to edit the sources repo using editors like vim, Nano under the dir. of /etc/apt/sources.list

## **APT VS APT-GET**

APT has a more user-friendly environment as compared to the APT-GET and APT comes pre-installed with Debian distribution systems

.

Now Let's see the difference between those two and why we have suggested APT

*lets install firebox using both APT and APT-GET*

```
[~]$ apt install firefox

Recommended packages:
  xul-ext-ubufox
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 36 not
upgraded.
Need to get 0 B/52.0 MB of archives.
After this operation, 202 MB of additional disk space will
be used.
Selecting previously unselected package firefox.
(Reading database ... 416280 files and directories currently
installed.)
Preparing to unpack
.../firefox_74.0+linuxmint2+tricia_amd64.deb ...
Unpacking firefox (74.0+linuxmint2+tricia) ...

Progress: [ 17%]
[#####.....]
.....]
```

*Here you can see the user friendly environment where there is a progress bar showing of how much downloaded has been completed . in this , only important information has been given , no rush .*

```
[~]$ apt-get install firefox
```

```
The following NEW packages will be installed:
```

```
firefox
```

```
0 upgraded, 1 newly installed, 0 to remove and 36 not  
upgraded.
```

```
Need to get 0 B/52.0 MB of archives.
```

```
After this operation, 202 MB of additional disk space will  
be used.
```

```
Selecting previously unselected package firefox.
```

```
(Reading database ... 416280 files and directories currently  
installed.)
```

```
Preparing to unpack
```

```
.../firefox_74.0+linuxmint2+tricia_amd64.deb ...
```

```
Unpacking firefox (74.0+linuxmint2+tricia) ...
```

```
Setting up firefox (74.0+linuxmint2+tricia) ...
```

```
Please restart all running instances of firefox, or you will  
experience problems.
```

```
Processing triggers for gnome-menus (3.13.3-11ubuntu1.1) ...
```

```
Processing triggers for hicolor-icon-theme (0.17-2) ...
```

```
Processing triggers for mime-support (3.60ubuntu1) ...
```

```
Processing triggers for desktop-file-utils (0.23+linuxmint8)
```

```
...
```

```
Processing triggers for mintsystem (8.4.6) ...
```

```
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
```

Here we have installed Firefox using APT-GET . so much of non-informative things are there , no user friendly environment.



WRITTEN BY Biswaraj Sahoo — AWS Community Builder | DevOps Engineer | Docker | Linux | Jenkins | AWS | Git | Terraform | Docker | kubernetes

Empowering communities via open source and education.

Connect with me over linktree: [linktr.ee/biswaraj333](https://linktr.ee/biswaraj333)

