RSync is a remote file (or data) synchronization protocol that enables you to *synchronize* a file stored on a local computer against a file stored on a remote computer - so that after synchronization the local and remote files are identical. If there are any differences between the local and remote file, RSync detects these differences and exchanges only the differences (+ merge instructions) between the local and remote computer, so the two files can be made identical.

RPM Command in Linux

RPM stands for Red Hat Package Manager. It is an open-source package manager (default) and the most famous utility of package management for Red Hat-based systems such as Fedora, CentOS, and RHEL. The tool permits system users and administrators for installing, updating, uninstalling, querying, verifying, and managing system software packages in [Linux/UNIX](https://www.javatpoint.com/linux-tutorial) operating systems.

Formerly, the RPM is called the .rpm file. It contains compiled software libraries and programs required by the packages. It only implements with those packages that were created in .rpm format.

A few facts about RPM

* ***RPM (Red Hat Package Manager)*** is free and published upon ***GPL (General Public License).***
* RPM is a single way for installing packages upon Linux systems. If we have installed any package with source code, the RPM would not manage it.
* RPM keeps the details of every installed package in a database, i.e., ***/var/lib/rpm.***
* RPM deals with all the ***.rpm*** files. It includes the actual details of the packages like ***what it is, version info, dependencies info, from where it comes,*** etc.

Modes of RPM Command

* **Install:** This mode is used for installing the RPM packages.
* **Remove:** It is used for erasing, removing, or uninstalling the RPM packages.
* **Upgrade:** It is used for updating the available RPM packages.
* **Verify:** This mode is used for verifying the RPM packages.
* **Query:** This mode is used for querying the RPM packages.

**Where to Search RPM Package**

The following is the rpm site's list where we can search and download the RPM packages.

* [http://rpmfind.net](http://rpmfind.net/)
* [http://www.redhat.com](https://www.redhat.com/en)
* [http://freshrpms.net/](https://freshrpms.net/)
* <http://rpm.pbone.net/>

Note: In Linux, please remember we must be a super user at the time of installing the RPM packages. We can manage the rpm commands using appropriate actions with root privileges.

1. Install RPM in Linux

We can install the RPM package using the following command:

* $ sudo apt install rpm



2. Check the RPM Signature Package

Before installing the packages on our Linux systems always check a PGP signature of them and ensure their origin and integrity are ***OK.*** We can use the below command using an option, i.e., ***-checksig*** (stands for ***check signature***) for checking the package's signature which is known as a ***apacheds-2.0.0.AM26-i386.rpm.***

* # rpm --checksig apacheds-2.0.0.AM26-i386.rpm



3. Check the RPM Package Dependency before installing

Let's assume we wish to check the dependency of the RPM package before upgrading or installing a package. **For example,** we can use the below command for checking the dependency of a package, i.e., ***apacheds-2.0.0.AM26-i386.rpm.*** It will show the package dependencies list:

* # rpm -qpR apacheds-2.0.0.AM26-i386.rpm



**Where,**

* **-q:** It is used for querying any package.
* **-p:** It is used for listing the capabilities that this package gives.
* **-R:** This option is used for listing the capabilities over which the package depends.

4. Install the RPM Package without the Dependencies

If we know that every needed package is already installed and the RPM is only being stupid, then we can avoid the dependencies with the help of the ***-nodeps*** option (means no dependency check) before installing any package.

The command is mentioned as follows:

* # rpm -ivh --nodeps apacheds-2.0.0.AM26-i386.rpm



Forcefully, the above command will install the RPM package by avoiding the errors of dependencies. However, when those files of dependency are missing, the program will not implement at all, until we install them.

5. Check the RPM Package (Installed)

Using an option -q along with the package name will display whether the RPM package is installed or not.

The command is as follows:

* # rpm -q apacheds-2.0.0.AM26-i386.rpm



6. List each file of the installed RPM package

We can use the -ql option (query list) with the RPM command for viewing each file of the installed RPM package.

The command is as follows:

* # rpm -ql apacheds-2.0.0.AM26-i386.rpm



7. List RPM Packages (Recently Installed)

We can apply the below command of RPM with an option, i.e., ***-qa (query all).*** This option will list every RPM package that is recently installed.

* # rpm -qa --last

The output will be as follows:



8. List Each RPM Package (Installed)

We can use the below command for printing each name of the package (installed) on our Linux system.

* # rpm -qa

the output is as follows:



9. Remove the RPM Package

For uninstalling or removing the RPM package, for instance, we can use the name of the package ***apacheds-2.0.0.AM26-i386.rpm,*** not the real name of the package ***apacheds-2.0.0.AM26-i386.rpm.*** In the following command, we are using an option, i.e., ***-e (erase)*** for removing the package.

* # rpm -evv apacheds-2.0.0.AM26-i386.rpm

The output will be as follows:



10. Query the Details of RPM Package (Installed)

Let's say we have installed any RPM package and we wish to know the details of the package. The below option, i.e., ***-qi (query info)*** will print the details of an installed package that are available.

* # rpm -qi apacheds-2.0.0.AM26-i386.rpm

The output is as follows:



11. Details of the RPM Package Before Installing

We have downloaded any package using the Internet and we wish to know the details of this package before installing. The below -qip (query info package) option will print the details about the package.

* # rpm -qip apacheds-2.0.0.AM26-i386.rpm

The output is as follows:



yum Command Examples in Linux

The yum command improves the functionality of rpm while still using .rpm packages and maintaining an RPM database. It provides a more straightforward method for managing packages. One of the biggest benefits of YUM is the ability to automatically handle software dependencies. This means that administrators can tell YUM to install a particular package, along with automatically installing any additional packages that the package depends on.

An additional YUM benefit is the use of repositories. Repositories are storage locations for .rpm files. Repositories enable administrators to more easily maintain version control over the software.

**Syntax**

The syntax of the yum command is:

# yum [options] [subcommand] [package name]

**The -y Option**

Use the -y option with the yum command to automatically answer yes to installing additional software dependencies. If you do not, YUM will prompt you to answer yes or no to whether the additional dependencies should be installed.

**yum Subcommands**

The yum command comes with several subcommands for managing packages.

|  |  |
| --- | --- |
| **Subcommand** | **Used To** |
| install {package name} | Install the package from any configured repository. |
| localinstall {package name} | Install package from local repository. |
| remove {package name} | Uninstall the package. |
| update [package name] | Update the package; if none provided, updates all installed packages (time-consuming). |
| info {package name} | Report information about the package. |
| provides {file name} | Report what package provides the specified files or libraries. |

yum Command Examples

**1. To Install**

a. To install any package on the Linux system, we can fire the yum install and the package name:

# yum install package1.rpm

# yum install package1.rpm package2.rpm package3.rpm package4.rpm

b. Yum utility generally ask for the confirmation for package installation, if you want to specify it in the command itself then fire below command

# yum install package1.rpm -y

**2. To Search**

To search for any package on the RPM repository (it can be RHN, Cent OS repository etc.):

# yum search package1.rpm

# yum search package1 package2

# yum search all

**3. To Update**

To update any existing package on the system fire below update command:

# yum update package1.rpm

# yum update package1.rpm package2.rpm

# yum update package1.rpm -y

**4. To remove/uninstall**

To remove any existing package from the system:

# yum remove package1.rpm

# yum remove package1.rpm package2.rpm

# yum remove package1.rpm -y

**5. To update**

To update the entire system for the available updates from Vendor repository:

# yum check-update

# yum update

# yum update –y

# yum update yum

**6. Get info**

To get the information about any package:

# yum info yum

# yum info vsftpd

**7. To see the lists**

To list packages we can fire below commands:

# yum list all

# yum list available

# yum list installed

# yum list extras

# yum list updates

# yum list obsoletes

# yum list recent

# yum list | less

# yum list pkgspec

**8. To clean**

To clean the downloaded packages or metadata or information cached by yum utility:

# yum clean package1.rpm

# yum clean metadata

# yum clean expire-cache

# yum clean rpmdb

# yum clean plugins

# yum clean all

**9. To see what provides**

To see what service/file is provided by which package:

# yum provides vsftpd

# yum provides yum

# yum whatprovides vsftpd

# yum whatprovides /etc/passwd

**10. For groups**

The packages are grouped into Group List, like DNS, Desktop, Web Server etc. So to view/install/update the packages from grouplist fire below command.

# yum grouplist

# yum groupinstall "DNS Name Server"

# yum groupinstall "DNS Name Server" "Graphical Internet"

# yum groupupdate "DNS Name Server"

# yum groupremove "DNS Name Server"

# yum groupremove "DNS Name Server" "Graphical Internet"

**11. Get info about group**

To get the information about the package groups:

# yum groupinfo "Graphical Internet"

# yum groupinfo "Graphical Internet" "DNS Name Server"

**12. Resolvedep**

To specify to resolve the dependencies if any while installing or update the packages:

# yum resolvedep vsftpd

**13. To download only**

To just download the packages on the system we can fire below commands:

# yum install yum-downloadonly

# yum install vsftpd --downloadonly

# yum update vsftpd --downloadonly

# yum update --downloadonly --downloaddir=/tmp

**14. To locally install**

To install the downloaded packages from any local directory:

# yum localinstall package1.rpm

# yum localinstall package1.rpm package2.rpm

# yum localupdate package1.rpm

# yum localupdate package1.rpm package2.rpm

**15. Reinstall**

To re-install any package, i.e. to unistall and then install it again:

# yum reinstall vsftpd

**16. Downgrade**

To downgrade any package to the previous version:

# yum downgrade vsftpd

# yum downgrade vsftpd bash

**17. To see dependency list**

To see any dependencies for the specified package:

# yum deplist vsftpd

# yum deplist vsftpd bash

**18. Version**

To see the version information about any package:

# yum version all

# yum version installed

# yum version available

# yum version group

# yum version grouplist

**19. History**

This allows the user to view what has happened in past transactions:

# yum history

# yum history info

# yum history list

# yum history summary

# yum history redo

# yum history undo

# yum history new