



Yum cheat sheet -- Comprehensive list of yum commands and flags



Team Packagecloud

📅 Mar 14, 2023 · 9 min read

[RPM-HOW-TO](#)

Yum is a command-line package manager for RPM-based Linux systems, such as Red Hat Enterprise Linux, CentOS, Fedora, and Oracle Linux. It stands for "Yellowdog Updater, Modified" and was originally developed by Yellowdog Linux as a way to easily manage packages on their distribution.

Yum makes it easy to search for, install, remove, and update packages and dependencies, as well as handle any conflicts that may arise. It uses repositories, which are online collections of packages that can be installed on your system. These repositories contain metadata about the packages, such as version numbers and dependencies, which yum uses to resolve dependencies and ensure that the correct versions of packages are installed. Unlike it's Debian

cousin, which uses [various commands for package management](#), yum can perform all package operations by itself.

Yum can also be used to manage system updates, including security updates and bug fixes. By regularly running yum update, you can ensure that your system is up-to-date and secure.

Useful flags

Assuming yes for all prompts

```
--y
```

Assuming no for all prompts

```
--assumeno
```

Disabling GPG verification

```
--nogpgcheck
```

Skipping broken packages

Skips packages that whose dependencies can't be resolved.

```
--skip-broken
```

Enabling/Disabling repository for single execution

You can dynamically enable or disable a repository for a single execution.

```
--disable-repo=[path]
```

```
--enable-repo=[path]
```

Quiet mode

Suppresses output except for errors and warnings.

```
-q
```

Changelog

Display the changelog of a package

```
--changelog
```

Package Management

Installing an RPM package

Installing an RPM package from remote repositories

```
$ sudo yum install [package-name]
```

Installing a local RPM package

```
$ sudo yum install /path/to/vim-1.2.rpm
```

Installing a specific version of a package

```
$ sudo yum install gcc-4.0
```

Removing an RPM package and dependencies

Removes a package and any package it depends on (provided nothing else depends on it).

```
$ sudo yum remove [package-name]
```

Note: this will only remove the binaries and libraries, any configuration files will stay intact.

Downgrading a package

This will install the package's previous version.

```
$ sudo yum downgrade [package-name]
```

Viewing a package's dependencies

```
$ sudo yum deplist [package-name]
```

Listing packages

The `yum list` command can take different arguments:

Listing all available packages from repositories

```
$ sudo yum list available
```

Listing installed packages

```
$ sudo yum list installed
```

Listing installed and available packages

```
$ sudo yum list all
```

Listing all packages (installed or available) that match a given [package-name], can be a glob

```
$ sudo yum list [package-name]
```

```
$ sudo yum list mysql*
```

Searching for package

This searches for [package-name] across all repositories, also looking inside package descriptions.

```
$ sudo yum search [package-name]
```

Upgrading all system packages

```
$ sudo yum upgrade
```

This command installs **all** of the latest versions of each package installed on the system and is, generally, not recommended to be run on production systems.

Reinstalling a single package

Sometimes, it's necessary to force reinstallation of a package.

```
$ sudo yum reinstall [package-name]
```

Viewing info for a package

```
$ sudo yum info [package-name]
```

Finding which RPM package installs a given file

This command is very handy when it's not obvious which package needs to be installed to use it.

```
$ sudo yum provides [file]
```

`yum provides` can also take a glob:

```
$ sudo yum provides "**/bin/vim"
```

Listing all dependencies for a given package

```
$ sudo yum provides [package-name]
```

Package groups

yum-groups-manager is a command-line tool used to manage package groups in yum, a package manager for RPM-based Linux systems. yum-groups-manager allows you to create, edit, and delete package groups, as well as modify the packages and dependencies included in each group. You can also use it to generate a new yum repository configuration file containing only the package groups you specify, which can be useful for creating custom repositories.

Package groups are collections of packages that can be installed together, typically for a specific purpose, such as a desktop environment or web server.

Listing all groups

```
$ sudo yum group list
```

Installing all packages for a group

```
$ sudo yum group install "Basic Web Server"
```

Removing all packages for a group

```
$ sudo yum group remove "Basic Web Server"
```

Repository Management

Listing all repositories

```
$ sudo yum repolist
```

Listing all packages for a given [repository]

```
$ sudo yum repo-pkgs [repository] list
```

Installing all packages from given [repository]

```
$ sudo yum repo-pkgs [repository] install
```

Removing all packages from a given [repository]

```
$ sudo yum repo-pkgs [repository] remove
```

Updating local metadata cache

This is run automatically by yum as needed, but can be refreshed manually with `yum makecache`

```
$ sudo yum makecache
```

When this command is run, all available packages are fetched and re-indexed from the repositories yum knows about.

yum-utils and yumdownloader

yum-utils is a collection of utilities and plugins for yum, a package manager for RPM-based Linux systems. These utilities and plugins are designed to extend the functionality of yum and provide additional tools for managing packages and repositories.

yumdownloader is a command-line utility that comes with the yum-utils package in RPM-based Linux systems. It allows you to download packages from yum repositories without installing them, which can be useful for creating local repositories or installing packages on systems without internet access.

Downloading RPMs

Downloading RPM from remote repositories

```
$ sudo yumdownloader [package-name]
```

Downloading Source RPMs

```
$ sudo yumdownloader --source [package-name]
```

Downloading all dependencies for an RPM

```
$ sudo yumdownloader --resolve [package-name]
```

Filtering by architecture

```
$ sudo yumdownloader --archlist=[arch-list] [package-name]
```

More

yum-utils adds other useful commands to `yum` that are too specific to cover here but are still worth looking over at the [YumUtils website](#).

You can find a complete list of yum commands [here](#).

DNF

[DNF](#) is the next generation of package management for redhat based operating systems.

DNF, or Dandified Yum, is a modern package management tool that has been developed as a successor to the popular Yum package manager for RPM-based Linux distributions. DNF is designed to provide better performance, improved dependency resolution, and a more user-friendly command-line interface compared to Yum.

Some of the key features and benefits of DNF include:

- **Faster and more efficient package management:** DNF is faster than Yum and uses less memory, making it more efficient for large-scale package management tasks.
- **Improved dependency resolution:** DNF uses a more advanced algorithm for dependency resolution, which can help prevent conflicts and ensure that packages are installed and updated correctly.
- **User-friendly command-line interface:** DNF has a more intuitive and user-friendly command-line interface, with features such as tab completion and easy-to-read output.
- **Built-in plugins:** DNF comes with a number of built-in plugins for tasks such as managing repositories, cleaning up old packages, and downloading packages without installing them.
- **Support for modular content:** DNF supports modular content, which allows packages to be grouped together in modules with different lifecycles and update policies.

Getting more familiar with your package manager's tools can help you be more productive when

finding, installing, and querying packages. There is no need to memorize or learn the commands, just use this cheat sheet!

You might also like other posts...



RPM-HOW-TO

May 01, 2023 · 6 min read

Inspecting and extracting RPM package contents with & without installing the package

Learn how to inspect and extract RPM package contents with working codes and examples in this article.



Joe Damato

[RPM-HOW-TO](#)

Sep 18, 2021 · 7 min read

What is RPM and how do I use it?

This article explains about RPM and how you can use it. Also, you can find the comparisons between RPM and Packagecloud.



Packagecloud Ops

Focus on your code and let us worry about package management

Get Started →



packagecloud

SUBSCRIBE TO OUR NEWSLETTER

Email Address

Subscribe

Company

Support

Package MGMT & Hosting

CI Integrations

Build Tool Integrations

Legal

● STATUS



© 2025 Packagecloud. All trademarks and registered marks belong to their respective owners.