

Subscribe to our blog via email

Sign up!



Subscribe to our RSS feed

[back to posts](#)

Yum Cheat Sheet

Apr 5, 2015 ▪ packagecloud

[rpm-howto](#)

What is yum?

`yum` is the default package manager for RPM-based linux distributions (CentOS, Fedora, RHEL, Oracle). It's written in python and it stands for "Yellowdog Updater, Modified", as it was originally called "yup", the package manager for [Yellow Dog Linux](#). Unlike it's debian cousin, which uses [various commands for package management](#), yum can perform all package operations by itself.

Create an RPM repository in less than 10 seconds, free.

[Sign up!](#)

Useful flags

Assume yes for all prompts

```
--y
```

Assume no for all prompts

```
--assumeno
```

Disable GPG verification

```
--nogpgcheck
```

Skip broken packages

Skips packages that whose dependencies can't be resolved.

```
--skip-broken
```

Enable/Disable repository

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

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

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

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.

Downgrade a package

This will install the package's previous version.

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

View a package's dependencies

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

Listing packages

The `yum list` command can take different arguments:

List all available packages from repositories

```
$ sudo yum list available
```

List installed packages

```
$ sudo yum list installed
```

List installed and available packages

```
$ sudo yum list all
```

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

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

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

Search for package

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

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

Upgrade 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.

Reinstall a single package

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

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

View info for a package

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

Find 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"
```


List all dependencies for a given package

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

Package Groups

*Note: yum now has a **groups** subcommand for group operations, versions before 3.4.x should refer to [this document](#) instead.*

yum has the concept of “package groups”, groups of related packages that can be installed or uninstalled at once, that don’t necessarily depend on each other.

List all groups

```
$ sudo yum group list
```

Install all packages for a group

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

Remove all packages for a group

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

Repository Management

List all repositories

```
$ sudo yum repolist
```

List all packages for a given [repository]

(Note: yum > 3.4.x only)

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

Install all packages from given [repository]

(Note: yum > 3.4.x only)

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

Remove all packages from a given [repository]

(Note: yum > 3.4.x only)

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

Update 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

In order to download source packages, it's necessary to install an additional package for yum, called "yum-utils", which provides a `yumdownloader` binary, among other things.

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](#).

The Future: DNF

DNF is the next generation of package management for redhat based operating systems. Currently, it exists as a fork of yum 3.4 that use **libsolv** as its dependency solver. It is currently in use by the **Fedora** distribution.

Conclusion

Getting more familiar with your package manager's tools can help you be more productive when finding, installing, and querying packages.

We highly recommend that users of production redhat based systems become familiar with **yum** and **dnf**.

You can learn more about the tools mentioned in this blog post by reading the man page: **man 8 yum**