

Fedora

- Paketmanager in Fedora DNF

Paketmanager in Fedora

DNF

Using the DNF software package manager

DNF is a software package manager that installs, updates, and removes packages on Fedora and is the successor to YUM (Yellow-Dog Updater Modified). DNF makes it easy to maintain packages by automatically checking for dependencies and determines the actions required to install packages. This method eliminates the need to manually install or update the package, and its dependencies, using the `rpm` command. DNF is now the default software package management tool in Fedora.

Usage

`dnf` can be used exactly as `yum` to search, install or remove packages.

To search the repositories for a package type:

```
# dnf search packagename
```

To install the package:

```
# dnf install packagename
```

To remove a package:

```
# dnf remove packagename
```

Other common DNF commands include:

- `autoremove` - removes packages installed as dependencies that are no longer required by currently installed programs.
- `check-update` - checks for updates, but does not download or install the packages.
- `downgrade` - reverts to the previous version of a package.
- `info` - provides basic information about the package including name, version, release, and description.
- `reinstall` - reinstalls the currently installed package.
- `upgrade` - checks the repositories for newer packages and updates them.
- `exclude` - exclude a package from the transaction.

For more DNF commands refer to the man pages by typing `man dnf` at the command-line, or [DNF](#)

[Read The Docs](#)

Automatic Updates

The `dnf-automatic` package is a component that allows automatic download and installation of updates. It can automatically monitor and report, via e-mail, the availability of updates or send a log about downloaded packages and installed updates.

For more information, refer to the [Read the Docs: DNF-Automatic](#) page.

System Upgrades

The Fedora system can be upgraded directly with DNF, or with the DNF system upgrade plugin.

Refer to the [DNF System Upgrade](#) document for more details.

Language Support Using DNF

DNF can be used to install or remove Language Support. A detailed description with a list of available languages can be found on [Language Support Using Dnf](#) page.

Plugins

The core DNF functionality can be extended with plugins. There are officially supported [Core DNF plugins](#) and also third-party [Extras DNF Plugins](#). To install them, run

```
# dnf install dnf-plugins-core-PLUGIN_NAME
```

or

```
# dnf install dnf-plugins-extras-PLUGIN_NAME
```

Excluding Packages From Transactions

Sometimes it is useful to ignore specific packages from transactions, such as updates. One such case, for example, could be when an update includes a regression or a bug. DNF allows you to exclude a package from the transaction:

- using the command line

```
sudo dnf upgrade --exclude=packagename
```

- using its configuration files

You can add a line to `/etc/dnf/dnf.conf` to exclude packages:

```
excludepkgs=packagename
```

This can also be added to the specific repository configuration files in `/etc/yum.repos.d/`. [Globs](#) may be used here to list multiple packages, and each specification must be separated by a comma. If you have used this configuration, you can disable it in individual DNF commands using the `--disableexcludes` command line switch.

If you use a GUI update application which does not allow you to specify packages to exclude when they run, this method can be used.

Using the DNF Versionlock plugin

You can also use the DNF `versionlock` plugin to limit the packages that are included in a transaction. It allows you to list what versions of particular packages should be considered in a transaction. All other versions of the specified packages will be ignored. The plugin is part of `dnf-plugins-core` package and can be installed using the command below:

```
sudo dnf install 'dnf-command(versionlock)'
```

To lock the currently installed version of a package, use:

```
sudo dnf versionlock add package
```

To remove the version lock, use:

```
sudo dnf versionlock delete package
```

The `list` command can be used to list all locked packages, while the `clear` command will delete all locked entries.

References

1. [DNF Command Reference](#)
2. [DNF wiki](#)
3. [DNF VersionLock](#)