



## Contents

Performing system upgrade

Optional post-upgrade tasks

Update system configuration files

Update GRUB bootloader on BIOS systems

Clean-up retired packages

Clean-up old packages

Clean-up old kernels

Clean-up old symlinks

Resolving post-upgrade issues

Rebuilding the RPM database

Using distro-sync to resolve dependency issues

Relabel files with the latest SELinux policy

Frequently Asked Questions

How do I report issues with the upgrade?

Does DNF System Upgrade verify the software it runs or installs during an upgrade?

Will packages in third-party repositories be upgraded?

Can I upgrade from an End-of-Life (EOL) release?

Can I do a single upgrade across many releases (i.e. 30→34)?

Can I use DNF System Upgrade to upgrade to a pre-release (e.g. a Beta)?

## Categories

Administration

How-To Upgrade



# Upgrading Fedora Using DNF System Plugin

Michael Wu, Anthony McGlone, The Fedora Docs team – Version F38, F39 – Last review: 2023-11-07

## Contents

Performing system upgrade

Optional post-upgrade tasks

Update system configuration files		
Update GRUB bootloader on BIOS systems		<input type="checkbox"/> In this section 
Clean-up retired packages		
Clean-up old packages		
Clean-up old kernels		
Clean-up old symlinks		
Resolving post-upgrade issues		
Rebuilding the RPM database		
Using distro-sync to resolve dependency issues		
Relabel files with the latest SELinux policy		
Frequently Asked Questions		
How do I report issues with the upgrade?		
Does DNF System Upgrade verify the software it runs or installs during an upgrade?		
Will packages in third-party repositories be upgraded?		
Can I upgrade from an End-of-Life (EOL) release?		
Can I do a single upgrade across many releases (i.e. 30→34)?		
Can I use DNF System Upgrade to upgrade to a pre-release (e.g. a Beta)?		

[dnf-plugin-system-upgrade](#) is a plugin for the [DNF](#) package manager and is used to upgrade your system to the current release of Fedora. For Fedora Silverblue and Fedora CoreOS, which use [rpm-ostree](#), you may refer to [rpm-ostree documentation](#) for details.

This is the recommended command-line upgrade method. It works as follows:

- Packages are downloaded while the system is running normally

- The system reboots into a special environment (implemented as a systemd target) to install them

- Upon completion, the system reboots into the new Fedora release

## Performing system upgrade

---



## Warning



In this section



**Back up your data** before performing a system-wide upgrade as every system upgrade is potentially risky. As a precaution, download the [Fedora Workstation Live image](#) in the event something goes wrong.

To update your Fedora release from the command-line do:

```
sudo dnf upgrade --refresh
```

and reboot your computer.

**Important:** Do not skip this step. System updates are required to receive signing keys of higher-versioned releases, and they often fix problems related to the upgrade process.

Install the `dnf-plugin-system-upgrade` package if it is not currently installed:

```
sudo dnf install dnf-plugin-system-upgrade
```

Download the updated packages:

```
sudo dnf system-upgrade download --releasever=39
```

Change the `--releasever=` number if you want to upgrade to a different release. Most people will want to upgrade to the latest stable release, which is `39`, but in some cases, such as when you're currently running an older release than `38`, you may want to upgrade just to Fedora `38`. System upgrade is only officially supported and tested over 2 releases at most (e.g. from `37` to `39`). If you need to upgrade over more releases, it is recommended to do it in several smaller steps ([read more](#)).

You can also use `40` to upgrade to a [Branched](#) release, or `rawhide` to upgrade to [Rawhide](#). Note that neither of these two are stable releases. For details about the upgrade process and common issues related to those two releases, please look at appropriate sections on [aforelinked pages](#).

If some of your packages have unsatisfied dependencies, the upgrade will refuse to continue until you run it again.    

This often happens with packages installed from third-party repositories for which an updated repository hasn't been yet published. Study the output very carefully and examine which packages are going to be removed. None of them should be essential for system functionality, but some of them might be important for your productivity.

- In case of unsatisfied dependencies, you can sometimes see more details if you add `--best` option to the command line.
- If you want to remove/install some packages manually before running `dnf system-upgrade download` again, it is advisable to perform those operations with `--setopt=keepcache=1` `dnf` command line option. Otherwise the whole package cache will be removed after your operation, and you will need to download all the packages once again.

When the new GPG key is imported, you are asked to verify the key's fingerprint. Refer to <https://getfedora.org/security> to do so.

Trigger the upgrade process. This will reboot your machine (immediately!, without a countdown or confirmation, so close other programs and save your work) into the upgrade process running in a console terminal:

```
sudo dnf system-upgrade reboot
```

Once the upgrade process completes, your system will reboot a second time into the updated release version of Fedora.

## Optional post-upgrade tasks

---

These are some of the tasks you can do after a successful upgrade.

### Update system configuration files

Most configuration files are stored in the `/etc` folder. If you have changed the package's configuration files, RPM creates new files with either `.rpmnew` (the new

default config file), or `.rpmsave` (your old config file backed up). You can search for these files, or use the `rpmconf` tool that ships with `rpmconf`.  
enter:

```
sudo dnf install rpmconf
```

Once the install is complete enter:

```
sudo rpmconf -a
```

## Note

Some third-party packages drop edited configuration files in `/etc/yum.repos.d/` and reverting these files to their original versions may disable updates for the software. Please remember to review configuration files in this directory carefully.

For more information you can refer to the man pages (`man rpmconf`).

If you use `rpmconf` to upgrade the system configuration files supplied with the upgraded packages then some configuration files may change. After the upgrade you should verify `/etc/ssh/sshd_config`, `/etc/nsswitch.conf`, `/etc/ntp.conf` and others are expected. For example, if OpenSSH is upgraded then `sshd_config` reverts to the default package configuration. The default package configuration **does not** enable public key authentication, and allows password authentication.

## Update GRUB bootloader on BIOS systems

Systems with the [BIOS firmware](#) have the GRUB RPM packages updated. However, the installed or embedded bootloader is never updated automatically. It is a good idea to update it between Fedora release versions.

Find the device node the `/boot/` directory is located on:

```
$ sudo mount | grep "/boot "  
/dev/sda4 on /boot type ext4 (rw,relatime,seclabel)
```

The device node is `/dev/sda4`. Reinstall the bootloader while specifying the device node without the number:

```
$ sudo grub2-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
```

## Clean-up retired packages

With every release, Fedora retires a few packages. There are various reasons; the packages become obsolete, they have a dead upstream, or the maintainer steps down. Fedora no longer distributes these packages; however, they are still on your system. These packages will not receive upgrades. It is highly recommended to remove them.

If you upgrade across one release (e.g. Fedora 38 to 39), run the following commands:

```
$ sudo dnf install remove-retired-packages
$ remove-retired-packages
```

If you upgrade across two releases (e.g. Fedora 37 to 39), you must supply the old release version to `remove-retired-packages`:

```
$ sudo dnf install remove-retired-packages
$ remove-retired-packages 37
```

### Note

Upgrades across **more** than two releases are not supported.

## Clean-up old packages

You can see a list of packages with broken dependencies by typing:

```
sudo dnf repoquery --unsatisfied
```

In this section



The list should be empty, but if this is not the case consider removing them as they are not likely to work.

You can see duplicate packages (packages with multiple versions installed) with:

```
sudo dnf repoquery --duplicates
```

And you can remove them with:

```
sudo dnf remove --duplicates
```

## Note

Run `sudo dnf upgrade` first, as this list is only valid if you have a fully updated system. Otherwise, you will see a list of installed packages that are no longer in the repositories because an update is available. This list may also contain packages installed from third-party repositories who may not have updated their repositories.

For packages from the official repositories, the latest version should be installed. However, some packages that are still on your system may no longer be in the repositories. To see a list of these packages do:

```
sudo dnf list extras
```

If you see a package you do not need, or use, you can remove it with:

```
sudo dnf remove $(sudo dnf repoquery --extras --exclude=kernel,kern
```

You can safely remove packages no longer in use with:

```
sudo dnf autoremove
```

## Warning



In this section



DNF decides that a package is no longer needed if you haven't explicitly asked to install it and nothing else requires it. However, that doesn't mean that the package is not useful or that you don't use it. **Only remove what you are sure you don't need.**

## Clean-up old kernels

After you boot into the latest kernel and test the system you can remove previous kernels. Old kernels remain even after `dnf autoremove` to avoid unintentional removals.

One of the easier ways to remove old kernels is with a script that retains the latest kernel. The script below works whenever Fedora updates a kernel, and does not depend upon a system upgrade.

```
#!/usr/bin/env bash

old_kernels=$(dnf repoquery --installonly --latest-limit=-1 -q)
if [ "${#old_kernels[@]}" -eq 0 ]; then
    echo "No old kernels found"
    exit 0
fi

if ! dnf remove "${old_kernels[@]}; then
    echo "Failed to remove old kernels"
    exit 1
fi


echo "Removed old kernels"
exit 0
```

## Clean-up old symlinks

There may be some dangling symlinks in the filesystem after an upgrade. You can clean the dangling links by installing the `symlinks` utility and deleting the old links.



```
sudo dnf install symlinks
```

🔍 |  In this section 

Once the utility is installed you can audit for broken symlinks like shown below. `-r` means recursive.

```
sudo symlinks -r /usr | grep dangling
```

After you verify the list of broken symlinks you can delete them like shown below. `-d` means delete.

```
sudo symlinks -r -d /usr
```

## Resolving post-upgrade issues

---

### Note

Only follow these steps if you encounter problems with your upgraded system.

### Rebuilding the RPM database

If you see warnings when working with RPM/DNF tools, your database might be corrupt. It is possible to rebuild it to see if resolves your issues. Always back up `/var/lib/rpm/` first. To rebuild the database, run:

```
sudo rpm --rebuilddb
```

### Using distro-sync to resolve dependency issues

The system upgrade tool uses `dnf distro-sync` by default. If your system is partly upgraded or you see some package dependency issues, try running another `distro-sync` manually to see if this fixes the problem. This will attempt to make your installed packages the same version in your currently enabled repositories, even if it

must downgrade some packages:



In this section



```
sudo dnf distro-sync
```

You can also use the `--allowerasing` option will remove packages with dependencies that can not be satisfied. Always review which packages will be removed before confirming this:

```
sudo dnf distro-sync --allowerasing
```

## Relabel files with the latest SELinux policy

If you encounter any warnings regarding policies with SELinux, some files may have incorrect SELinux permissions. This may happen if SELinux was disabled in the past. To relabel SELinux on the system, run the following command and then reboot:

```
sudo fixfiles -B onboot
```

The boot process will likely take a long time, as it checks and fixes SELinux permission labels on all files in your system.

# Frequently Asked Questions

---

## How do I report issues with the upgrade?

See [Common bugs](#) to check if it is a known problem the community is already aware of.

Search [Bugzilla for an existing bug report](#) filed against the system-upgrade plug-in.

If you do not see a report that matches your symptoms, you can file a new report from the search page. Please follow the bug reporting instructions mentioned in the [README from the GitHub repo](#) or in `man dnf.plugin.system-upgrade`.

If you encounter any issues after the upgrade with a specific package, file a bug against the package with which you are having issues.

In this section



## Does DNF System Upgrade verify the software it runs or installs during an upgrade?

Yes. The package signing keys for the newer Fedora release are sent to older Fedora releases to allow DNF to verify the integrity of the downloaded packages. You can disable this function if needed, but is not recommended as you will be open to attacks from malicious software.

## Will packages in third-party repositories be upgraded?

Yes, if they are configured like regular DNF repositories and the version numbers are not hard-coded in the repository file (usually found in `/etc/yum.repos.d/`). Commonly used third-party repositories like RPM Fusion should work. However, if attempting to upgrade prior to, or soon after, an official Fedora release, they may not have updated their repository paths, and DNF may be unable to find their packages. Usually, this should not prevent the upgrade from running successfully. Also, you can update packages from the third-party repository later.

## Can I upgrade from an End-of-Life (EOL) release?

It is strongly recommended to upgrade an EOL release on any production system, or any system connected to the public internet.

Any upgrade from Fedora 20 or earlier is done **at your own risk** as DNF was not the default package management tool. However, if you do have a release newer than Fedora 20 that is EOL, you can attempt to do an upgrade, but this method is **not supported**. You may try to upgrade through intermediate releases until you reach a currently-supported release, or try to upgrade to a currently-supported release in a single operation. Again this is unsupported and is **at your own risk**.

## Can I do a single upgrade across many releases (i.e. 30→34)?

Upgrades to the very next release (e.g. 38 to 39) as well as upgrades skipping one

release (e.g. 37 to 39) are both supported. However, it is highly recommended to perform the upgrade before your release reaches End of Life (EOL), which is roughly a month after N+2 release has been released (when you're currently on release N). The [Fedora Release Life Cycle](#) is specifically designed to provide this approximate one month "grace period" to allow users the choice to upgrade their systems on a yearly basis, i.e. once every two releases. You can study [Releases](#) to see the current release status and schedule. Around a month after the new release comes out, the last-but-one release becomes End of Life (EOL). The upgrade is likely to work successfully after the release goes EOL, but the time period after the new release may be uncertain.

Upgrades across more than two releases are **not supported**, and issues encountered with such upgrades may not be considered significant bugs.

When upgrading across more than two releases, you may need to import the GPG key for the release you want to update to. You can do this with:

```
sudo rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-N-primary
```

(where N is the Fedora version.)

## Can I use DNF System Upgrade to upgrade to a pre-release (e.g. a Beta)?

Yes, but this is subject to temporary breakage as with any other aspect of a pre-release.

Want to help? [Learn how to contribute to Fedora Docs](#) >

All Fedora Documentation content available under CC BY-SA 4.0 or, when specifically noted, under another accepted free and open content license.





|  In this section



[Privacy Statement](#) [Legal](#) [Code of Conduct](#) [Sponsors](#)

Last build: 2023-11-27 21:50:09 UTC | Last content update: 2023-11-20



Sponsored by  
Red Hat

Fedora is sponsored by Red Hat.  
Learn more about the relationship between Red Hat and Fedora.