

Drivers

- [Nvidia Driver on Fedora](#)

Nvidia Driver on Fedora

NVIDIA Fedora Driver Guide

A little guide to help you install & manage the NVIDIA GPU driver on your Fedora system(s)

I am personally a **Fedora 40** user at the moment, so this is mostly what this guide applies to (though I believe it should work alright on newer releases, and also on older releases which are not old very old [something like Fedora 38+])

? Kindly note

that this guide explicates the procedure of installing & uninstalling the driver through the official Nvidia driver installer downloaded from nvidia.com.

If you would like to install the driver using the RPMFusion repository instead, please refer to <https://rpmfusion.org/Howto/NVIDIA>

Index of content

- [Driver installation](#)
- [Driver uninstallation](#)
- [Issues faced after installing the NVIDIA drivers, and how to solve them](#)
 - [Fix VLC black screen issue](#)
 - [\(for GNOME\) Wayland is no longer enabled/not visible on the login screen](#)
 - [Fix Wayland issues \(flickering, etc.\)](#)
- [References](#)

“

?? Warning

Please follow & read every part of this guide with fine care to avoid the occurrence of any problems.

Also do not worry if the system looks stuck during any rebooting step. It actually is not stuck! Kindly allow up to 2 minutes for the rebooting to complete.

Driver installation

1. Ensure that you have uninstalled any previously installed NVIDIA drivers:
 - to uninstall any Nvidia drivers installed from the RPMFusion repository:

```
sudo dnf remove xorg-x11-drv-nvidia\*
reboot
```

- to uninstall any Nvidia drivers installed using this guide: [Driver uninstallation](#)

2. Upgrade the system packages and install the required dependencies:

```
sudo dnf upgrade
sudo dnf install kernel-devel kernel-headers gcc make dkms acpid libglvnd-glx libglvnd-opengl
libglvnd-devel pkgconfig
```

3. Navigate to <https://www.nvidia.com/Download/index.aspx?lang=en-us> and download the proper driver for your GPU and Linux architecture. The website should give you a file that ends with the `.run` file extension.
4. Switch to the terminal view of your system by pressing `Alt + Ctrl + F3` (if this does not switch from the GUI mode to the terminal mode for you, try `Alt + Ctrl + F1` or `Alt + Ctrl + F2` instead for a different tty)

NOTE: Something that freaked me out on the Fedora terminal tty is that using the backspace key when there's nothing to erase produces a loud beep sound. You may need to pay attention to this `⏏`

5. Stop the GDM service:

```
sudo systemctl stop gdm
```

or if you are using the **Fedora KDE** spin, stop the SDDM service:

```
sudo systemctl stop sddm
```

Kindly note that it is important to stop the display manager (GDM/SDDM) service throughout the driver installation/uninstallation process as it may cause trouble otherwise.

6. Change to the path of the directory that includes the downloaded `.run` file using `cd`
7. Run the installer:

```
chmod +x NVIDIA-Linux-x86_64-555.42.02.run
sudo sh ./NVIDIA-Linux-x86_64-555.42.02.run
```

(make sure to replace the file name with the actual one that you got from the Nvidia website)

8. The installer will guide you through everything. Please read everything with care and answer the prompts depending on the proper situation to avoid any problems.

NOTE: If the installer asks you to disable Nouveau, allow the installer to disable it for you. You may need to abort the installer after this, then run `sudo dracut --regenerate-all --force && reboot`, then start again from step 4 once the system has completed rebooting.

9. Once the installer has completed installing the driver, run `sudo dracut --regenerate-all --force` to update the initramfs.
10. Edit `/etc/default/grub` using `sudo nano /etc/default/grub`
11. Add `nvidia-drm.modeset=1` and `nvidia-drm.fbdev=1` inside your `GRUB_CMDLINE_LINUX` (i.e. `GRUB_CMDLINE_LINUX="nvidia-drm.modeset=1 nvidia-drm.fbdev=1"`)
12. Run `sudo grub2-mkconfig -o /etc/grub2.cfg`
13. Reboot the system
14. Your newly installed driver should be up and running once the system boots up (you may run `nvidia-smi` to confirm so).
15. In order to enable video acceleration support, you need to install these packages: `sudo dnf install nvidia-vaapi-driver libva-utils vdpauinfo` (make sure to reboot the system after installing these packages)

Driver uninstallation

1. To ensure that we can boot into the system graphically through the Nouveau driver after uninstalling the Nvidia driver, remove any Nouveau-blacklist entries that might have been created by the installer previously:

```
sudo rm -rf /lib/modprobe.d/nvidia-installer-*
sudo rm -rf /etc/modprobe.d/nvidia-installer-*
sudo rm -rf /usr/lib/modprobe.d/nvidia-installer-*
```

2. Remove any entries related to the NVIDIA driver (`nvidia-drm.modeset`, `nvidia-drm.fbdev`, etc) from your `/etc/default/grub` file. **(this is important)**.
3. Rebuild the GRUB configuration using `sudo grub2-mkconfig -o /etc/grub2.cfg`
4. Uninstall any installed NVIDIA video acceleration packages:

```
sudo dnf remove nvidia-vaapi-driver libva-utils vdpauinfo
```

5. Run the uninstaller:

```
sudo nvidia-installer --uninstall
```

6. Run the below commands to restore everything to how it was before the installation of the driver:

```
sudo rm -f /usr/lib{,64}/libGL.so.* /usr/lib{,64}/libEGL.so.*  
sudo rm -f /usr/lib{,64}/xorg/modules/extensions/libglx.so  
sudo dnf reinstall xorg-x11-server-Xorg mesa-libGL mesa-libEGL libglvnd*
```

7. Rebuild the system initramfs:

```
sudo dracut --regenerate-all --force
```

8. Reboot the system once the uninstalling process has finished.

Issues faced after installing the NVIDIA drivers, and how to solve them

Getting a black screen on video players (VLC, etc)

This may happen because most of the video/media players require openh264 & a few other codecs which are not shipped by default with Fedora due to licensing reasons.

To solve this:

```
sudo dnf config-manager --enable fedora-cisco-openh264 -y  
sudo dnf install openh264 mozilla-openh264 libavcodec-freeworld ffmpeg mpv vlc gstreamer1-  
plugins-bad-freeworld gstreamer1-plugins-ugly
```

You may need to [have the RPMFusion free and nonfree repos enabled](#) for some of the above packages.

See <https://discussion.fedoraproject.org/t/cant-play-videos-in-firefox/79645/25> and <https://discussion.fedoraproject.org/t/codecs-missing-failing-to-play-h-264/104643/2> for more information on this matter.

(on GNOME) Wayland is not shown as an option on the login screen (or the cog icon of the login screen doesn't show at all)

1. Edit the `/etc/gdm/custom.conf` file using `sudo nano /etc/gdm/custom.conf`
2. Ensure that `WaylandEnable=true` is set in that file and make sure that it's uncommented (does not start with a `#`)
3. Run `sudo ln -s /dev/null /etc/udev/rules.d/61-gdm.rules`
4. Reboot the system

The above doesn't apply to a Fedora KDE spin installation since it enforces the use of Wayland and does not have X11 anyway.

The experience on Wayland is not the smoothest (fix Wayland issues)

This may happen for a lot of reasons. For a while now, NVIDIA has been known to have issues with the Wayland windowing system. However, NVIDIA has been working on making this better. And this has actually already gotten much better starting from the NVIDIA driver 555.42.02 which added [explicit sync](#) support.

So first of all, make sure to have:

- Version 555.42.02 or a higher version of the Nvidia driver
- GNOME 46.1+ (or KDE Plasma 6.1+) on your Fedora installation

then continue reading below to make the experience even smoother:

- Your system may be using the Mesa driver instead of the NVIDIA one on Wayland sessions. You can confirm this by typing `glxinfo|egrep "OpenGL vendor|OpenGL renderer*"`
In order to solve this:
 1. Edit `/etc/default/grub` using `sudo nano /etc/default/grub`
 2. Add `nvidia-drm.modeset=1` and `nvidia-drm.fbdev=1` inside your `GRUB_CMDLINE_LINUX` (i.e. `GRUB_CMDLINE_LINUX="nvidia-drm.modeset=1 nvidia-drm.fbdev=1"`)
 3. Run `sudo grub2-mkconfig -o /etc/grub2.cfg`
 4. Reboot the system
- You may have the GSP firmware of Nvidia enabled, and this is known to cause some performance issues ([especially on KDE Plasma](#)) on the beta 555.42.02 version of the

driver. Maybe this will be fixed in the future, but for now, we can disable the GSP firmware if needed.

You can check whether the GSP firmware is enabled or no by typing `nvidia-smi -q | grep "GSP Firmware"` — if it says `N/A` then the firmware is not enabled. If otherwise (it shows a version for GSP firmware) then the firmware is enabled.

To disable the GSP firmware, please follow the below steps:

1. Edit `/etc/default/grub` using `sudo nano /etc/default/grub`
2. Add `nvidia.NVreg_EnableGpuFirmware=0` inside your `GRUB_CMDLINE_LINUX`
3. Run `sudo grub2-mkconfig -o /etc/grub2.cfg`
4. Reboot the system

See <https://forums.developer.nvidia.com/t/major-kde-plasma-desktop-frameskip-lag-issues-on-driver-555/293606> for more information on this issue.

- for Google Chrome (and Chromium-based browsers in general), you may need to switch the "Preferred Ozone platform" flag to "Wayland" or "auto" (this is not needed if you have both GNOME 46.1+/KDE Plasma 6.1+ and Nvidia driver 555.42.02+). Follow the steps below in order to apply this:
 1. Go to `chrome://flags`
 2. Search "Preferred Ozone platform"
 3. Set the flag to "Wayland" or "auto"
 4. Restart the browser
- for some Electron apps, you may need to pass the same Ozone platform flag as we did above. For example `code --enable-features=UseOzonePlatform,WaylandWindowDecorations --ozone-platform-hint=auto` for Visual Studio Code
- You may not have the preserve video memory allocations module parameter enabled, and this can cause issues particularly when suspending and resuming the system, usually in the form of graphical artifacts or a broken desktop environment.

You can check whether the module parameter is enabled or not by typing `sudo cat /proc/driver/nvidia/params | grep "PreserveVideoMemoryAllocations"`. If the value is `0` or missing, then the parameter is not enabled.

To enable the preserve video memory allocations module parameter, please follow the below steps:

1. Edit `/etc/default/grub` using `sudo nano /etc/default/grub`
2. Add `nvidia.NVreg_PreserveVideoMemoryAllocations=1` inside your `GRUB_CMDLINE_LINUX`
3. Run `sudo grub2-mkconfig -o /etc/grub2.cfg`
4. Reboot the system
5. Run `sudo cat /proc/driver/nvidia/params | grep "PreserveVideoMemoryAllocations"` to verify the parameter is now set

If you are still experiencing issues with suspend/resume after enabling this module parameter, you may want to take a look at Nvidia's [power management documentation](#) to double check that the relevant `systemd` services are installed and enabled.
