Graphics Cards

Most graphical user interfaces these days are hardware accelerated, so the appropriate graphics driver will be needed for optimal performance and a smooth desktop experience. Additionally, these drivers provide 3D acceleration and hardware video decoding/encoding capabilities.

The Linux graphics stack consists of several components, but the main component is the `mesa` package.

Manufacturer	OpenGL	Vulkan	Video Acceleration
Intel	`mesa`	`vulkan-intel`	(see note below)
AMD	`mesa`	`vulkan-radeon`	part of `mesa`
NVIDIA (Proprietary)	`mesa`	`nvidia-utils`	`nvidia-utils`, `nvidia-vaapi-driver`
NVIDIA (Nouveau)	`mesa`	`vulkan-nouveau`	part of `mesa`

Intel

For modern Intel integrated graphics (Broadwell generation / 5th Gen Core and newer) and Intel Arc, install the following packages:

```
pacman -S mesa vulkan-intel
```

Hardware acceleration for video playback depends on the generation of Intel hardware:

- `intel-media-driver`: If your CPU is **5th gen or newer** (Broadwell, Skylake, Kaby Lake, up to modern Alder Lake/Raptor Lake and Intel Arc).
- `libva-intel-driver`: If your CPU is **4th gen or older** (Haswell, Ivy Bridge, Sandy Bridge).

AMDGPU

For modern AMD integrated and dedicated graphics cards (RDNA architectures and modern Vega/GCN configurations), install the following packages:

```
pacman -S mesa vulkan-radeon
```

NVIDIA

In the case of NVIDIA, there's the option to either use the open-source Nouveau drivers, or the official Linux kernel modules provided by NVIDIA themselves.

If you have a relatively recent NVIDIA card, it is generally recommended to go with the official NVIDIA drivers for optimal performance. For older cards or strictly open-source setups, use the Nouveau/Mesa stack.

Nouveau open-source driver

The open-source NVIDIA stack uses Nouveau:

> **ATTENTION:** For older cards (Maxwell/Pascal), dynamic re-clocking requires manual firmware extraction via the AUR package `nouveau-fw`. For newer Turing cards (RTX 20xx / GTX 16xx) and above, the required GSP firmware is natively included in the `linux-firmware` package.

```
pacman -S mesa vulkan-nouveau
```

Official NVIDIA driver

Modern Cards (RTX 20xx onwards)

Upstream and Arch Linux officially recommend the **open-source kernel modules** for all modern architectures.

```
pacman -S nvidia-open nvidia-vaapi-driver
```

Legacy Cards (GTX 10xx and earlier)

If you are running older supported hardware or need the closed-source driver, you must use the proprietary DKMS package available via the AUR:

```
yay -S nvidia-580xx-dkms nvidia-vaapi-driver
```

Early KMS Loading

To ensure the graphics drivers load as early as possible during the boot sequence (preventing display manager race conditions), add the modules to your initramfs configuration.

Make sure the following is included in the `MODULES` array of your `/etc/mkinitcpio.conf` file (or a drop-in config file under `/etc/mkinitcpio.conf.d/`):

> **WARNING:** Early loading these kernel modules into the initramfs will break system hibernation features by default because video memory preservation is enabled.

```
MODULES=(nvidia nvidia_modeset nvidia_uvm nvidia_drm)
```

Enable Kernel Mode Setting

Since `nvidia-utils` version 560+, Kernel Mode Setting (KMS) is enabled by default. However, when using older legacy drivers, KMS must be explicitly enabled through a kernel command-line argument at boot time to ensure Wayland compositors function properly:

```
nvidia_drm.modeset=1
```

To verify that kernel mode setting is active, query sysfs:

```
cat /sys/module/nvidia_drm/parameters/modeset
```

`Y` means Kernel Mode Setting was enabled on boot.
`N` means Kernel Mode Setting was **not** enabled on boot.

Revision #11
Created 2022-02-11 19:37:01 UTC by Sebin
Updated 2026-06-28 13:19:56 UTC by Sebin