

Installing Direct Rendering Manager drivers on FreeBSD 15.1 and enabeling Wayland with compositors.

Presently (2026-09-28) there are challenges with the setup of the FreeBDS online repositories that is making it rather cumbersome for novice users of FreeBSD to achive a working graphical Desktop Environment such as Plasma6/KDE6/Wayland.

The Issues are largely constrictet to FreeBSD 15.1, Installing on Freebsd 16-current is much more effortless.

As Installing Graphical drivers from online repos is almost impossible, it is required to Build and install the driver binaries using the PORTS Distribution. (/usr/ports/ directories)

The Ports Distribution on the 15.1 Release Media is now outdated as its from month of May 2026. Therefor it must not be used. Note: that the /usr/ports directory MUST be emptied of files before downloading the new PORTS distribution.

Login on the ROOT account.

If using UFS filesystem:

```
# cd /usr/ports; pwd      (verify that you are in directory /usr/ports before issueing rm -r command)
# rm -r *
```

If using ZFS filesystem

```
# umount /usr/ports
# zfs destroy zroot/usr/ports
# zfs create -o mountpoint=/usr/ports zroot/usr/ports
```

The procedure to get hold of an Updated PORTS distribution is described in the FreeBSD Handbook chapter 4.5.1 <https://docs.freebsd.org/en/books/handbook/ports/#ports-using-installation-methods> .Follow steps 1-2 do not implement step 3,4,5. You want the latest HEAD branch.

Copy file /usr/share/examples/etc/make.conf to /etc/make.conf

Edit the make.conf file at line 55: and select/type the CPUTYPE?= that is installed in your computer. (valid choises are listed in the file)

Run:

```
# pkg info| egrep "drm|nvidia|sddm"
```

If this command lists any already installed packages. You need to delete them as you are about to build them again for FreeBSD 15.1

```
# pkg delete 'package-name'
```

Install graphical desktops packages from repository :

```
# pkg install xorg wayland seatd gnome kde mate      ( takes a while )
```

This completes the preparations.

You should now have a up-to-date PORTS distribution installed and the Compiler is prepped to build for your machinetype.

Building the Direct Rendering Manager drivers.

Login as the ROOT account.

Its assumed that this is a newly installed & clean /usr/ports directory.

If it is a previously used /usr/ports there may be nessecity to clean the directory.

You are now expected to build the graphic DRM drivers. Proceed as follows:

```
# cd /usr/ports/graphics/drm-kmod
# make clean      ( if drm.kmod was previously built )
# make reinstall
```

After completion you will have packages drm-kmod and drm-612-kmod installed that populates /boot/modules with INTEL i915kms and AmdGPU drivers.

Note: add these lines to /etc/rc.conf to enable your GPU

For amdgpu: kld_list="amdgpu"

For Intel: kld_list="i915kms"

For radeonkms: kld_list="radeonkms"

Please ensure that all users requiring graphics are members of the "video" group.

Check /etc/group file, Add usernames to Video group with an editor as needed.

Reboot.

Login as user.

Executing command

```
$ startx
```

Should now start Basic windowmanger "twm" with three Xterm and Xclock.

Building NVIDIA drivers with Direct Rendering

Note: add this line to /etc/rc.conf to enable Nvidia graphics build

```
linux_enable="YES"
```

Reboot and confirm that linux.ko and linux64.ko are active kernel drivers with command

```
# kldstat
```

Then build the Nvidia drivers:

```
# cd /usr/ports/graphics/nvidia-drm-kmod
# make clean      (if nvidia-drm.kmod was previously built)
# make reinstall
```

This will install packages **nvidia-drm-kmod**, **nvidia-drm-612-kmod**, **nvidia-kmod**, **nvidia-driver**, that's Nvidia drivers for DRM and traditional Nvidia.

In addition you would want to also build Packages in directories

/usr/ports/x11/nvidia-settings	(GUI for NVIDIA card)
/usr/ports/x11/nvidia-xconfig	(Nvidia X11 configuration generator)
/usr/ports/x11/linux-nvidia-libs	(LIBS for linux Apps)

Note: add this lines to /boot/loader.conf

```
hw.nvidiadrmmodeset=1
hw.nvidia.registry.EnableGpuFirmware=1  (only required for Blackwell GPUs)
```

Note: add this line to /etc/rc.conf to enable your GPU

```
kld_list="nvidia-drm"
```

Please ensure that all users requiring graphics are members of the "video" group. Check /etc/group file, Add usernames to Video group with an editor as needed.

Reboot:

Login as user.

Executing command

```
$ startx
```

Should now start Basic windowmanger "twm" with three Xterm and Xclock.

Building SDDM Login manager

Due to a bug in SDDM its required to run the latest version of SDDM to be able to startup Wayland. The latest version is 0.21.0.36_6 . This version is not available in the online repos.

To build SDDM :

```
# cd /usr/ports/x11/sddm
# make clean ( if sddm was buildt previously)
# make reinstall
```

Add the following two lines to /etc/rc.conf :

```
seatd_enable="YES"
sddm_enable="YES"
```

Reboot system.

SDDM Login screen should appear.

NOTE: There is a menu of Desktop environments in the bottom Left corner of the screen.

Start with selecting Plasma6/X11 and try to login.

At the next session you can try Plasma6/Wayland.

Cursor issues with WAYLAND.

Wayland has issues with displaying the cursor.

The most common fix is to include these two environment variables in the users ~/.profile and logout and login again.

```
export WLR_NO_HARDWARE_CURSORS=1
export KWIN_FORCE_SW_CURSOR=1
```

Extra tasks.

For a full KDE experience installation of several additional packages can be done.

```
# pkg install kde-gear kdesdk kdeaccessibility kdevelop calligra
```

and ofcourse:

```
# pkg install libreoffice
```