

README for XFree86 4.0 on NetBSD

Rich Murphey, David Dawes, Marc Wandschneider, Mark Weaver, Matthieu Herrb

Last modified on: 20 February 2000

1. What and Where is XFree86?

XFree86 is a port of X11R6.4 that supports several versions of Intel-based Unix. It is derived from X386 1.2, which was the X server distributed with X11R5. This release consists of many new features and performance improvements as well as many bug fixes.

See the Copyright Notice.

The sources for XFree86 are available by anonymous ftp from:

<ftp://ftp.XFree86.org/pub/XFree86/4.0>

Binaries for NetBSD 1.3 and later are available from:
<ftp://ftp.XFree86.org/pub/XFree86/4.0/binaries/NetBSD>

A list of mirror sites is provided by <ftp://ftp.XFree86.org/pub/XFree86/MIRRORS>

XFree86 also builds on other NetBSD architectures. See section *Building on other architectures* (section , page) for details.

XFree86 also supports NetBSD on PC98 machines.

2. Bug Reports for This Document

Send email to <herrb@XFree86.Org> (Matthieu Herrb) or <XFree86@XFree86.org> if you have comments or suggestions about this file and we'll revise it.

3. New OS dependent features

See the Release Notes for non-OS dependent new features in XFree86 4.0.

3.1 New OS dependent features in 4.0

- Preliminary APM support.

3.2 New OS dependent features in 3.9.18

- Soft-booting secondary cards through the int10 BIOS interface is now possible using the x86emu real mode emulator.

3.3 New OS dependent features in 3.9.17

- Support for *silken mouse* with the wsmouse protocol has been added.

- A new version of the XFree86 Aperture driver which provides MTRR support is included.

4. Installing the Binaries

Refer to the Installation Document for detailed installation instructions.

5. Configuring X for Your Hardware

The `/etc/XF86Config` file tells the X server what kind of monitor, video card and mouse you have. You *must* create it to tell the server what specific hardware you have.

You'll need info on your hardware:

- Your mouse type, baud rate and its `/dev` entry.
- The video card's chipset (e.g. ET4000, S3, etc).
- Your monitor's sync frequencies.

For details about the `XF86Config` file format, refer to the *XF86Config(5)* manual page.

Once you've set up a `XF86Config` file, you can fine tune the video modes with the `xvidtune` utility.

5.1 About mouse configuration

The NetBSD `pms` mouse driver handles PS/2 style mice as `Busmouse`. Specify the protocol as "**busmouse**" in the mouse section of your `XF86Config` file if you're using a PS/2 mouse.

Only standard PS/2 mice are supported by this driver. Newest PS/2 mice that send more than three bytes at a time (especially Intellimouse, or MouseMan+ with a "3D" roller) are not supported yet.

XFree86 4.0 also has support for the mouse driver included in the new **wscons** console driver introduced by NetBSD 1.4. Specify "`wsmouse`" as the protocol and `/dev/wsmouse0`" as the device in `/etc/XF86Config` if you're using NetBSD 1.4 with `wscons`.

See `README.mouse` for general instruction on mouse configuration in XFree86.

6. Running X

The easiest way for new users to start X windows is to type:

```
startx >& startx.log
```

Error messages are lost unless you redirect them because the server takes over the screen.

To get out of X windows, type: "`exit`" in the console `xterm`. You can customize your X by creating `.xinitrc`, `.xserverrc`, and `.twmrc` files in your home directory as described in the `xinit` and `startx` man pages.

6.1 Starting Xdm, the display manager

To start the display manager, log in as root on the console and type: "`xdm -nodaemon`".

You can start `xdm` automatically on bootup by changing the line

```
xdm=NO                                xdm_flags=""                # x11 display manager
```

to:

```
xdm=YES                                xdm_flags=""                # x11 display manager
```

in `/etc/rc.conf`.

Under NetBSD 1.4 with the `wscons` console driver, you must enable a virtual console for the X server first. To do this follow these steps:

- Make sure the device file exists. If not, `"cd /dev ; ./MAKEDEV wscons"`.
- Next, make sure your kernel wants to do `wscons`. (see *below* (section , page)).
- Next, make sure `"wscons=YES"` in `/etc/rc.conf`.
- Next, make sure `/etc/wscons.conf` exists. The relevant bits:

```
#screen 0      -      vt100
screen 1      -      vt100
screen 2      -      vt100
screen 3      -      vt100
screen 4      -      -
screen 5      -      vt100
```

(Thanks to Mason Loring Bliss <mason@acheron.middleboro.ma.us> for this explanation)

Note that the binary distributions of XFree86 for NetBSD don't include support for the XDM-AUTHORIZATION-1 protocol.

7. Kernel Support for X

To make sure X support is enabled under NetBSD, the following line must be in your config file in `/sys/arch/i386/conf`:

```
options XSERVER, UCONSOLE
```

7.1 Console drivers

The server supports the standard NetBSD/i386 console drivers: `pccons`, `pcvt` and `wscons` (in `pcvt` compatibility mode). They are detected at runtime and no configuration of the server itself is required.

The `pccons` driver is the most widely tested and is the console driver contained in the NetBSD binary distribution's kernels.

The `pcvt` console driver is bundled with NetBSD. The `pcvt` X mode is compatible with the `pccons` driver X mode. It offers several virtual consoles and international keyboard support. In order to use this driver, change the line:

```
device          pc0      at isa? port "IO_KBD" irq 1
```

to

```
device          vt0      at isa? port "IO_KBD" irq 1
```

in your kernel config file, and rebuild and install your kernel.

XFree86 will also run with the `wscons` console driver in NetBSD 1.4. For now, it uses the `pcvt` compatibility mode, so be sure to have the lines:

```
options          WSDISPLAY_COMPAT_PCVT      # emulate some ioctls
options          WSDISPLAY_COMPAT_SYSCONS   # emulate some ioctls
options          WSDISPLAY_COMPAT_USL       # VT handling
options          WSDISPLAY_COMPAT_RAWKBD    # can get raw scancodes
```

in your kernel configuration file if you're using `wscons`. Refer to the `wscons(4)` and `wsmouse(4)` manual pages for informations on how to configure `wscons` into the kernel.

7.2 Aperture Driver

By default NetBSD include the BSD 4.4 kernel security feature that disable access to the `/dev/mem` device when in multi-users mode. But XFree86 servers can take advantage (or require) linear access to the display memory.

Most XFree86 4.0 card drivers require linear memory access. There are two ways to allow XFree86 to access linear memory:

The first way is to disable the kernel security feature by adding “`option INSECURE`” in the kernel configuration file and build a new kernel.

The second way is to install the aperture driver, included in source form in `xc/programs/Xserver/hw/xfree86/etc/apNetBSD.shar` in the XFree86 4.0 source distribution. Unpack it in a new directory of your choice by running:

```
sh apNetBSD.shar
```

By default the aperture driver will be installed in `/usr/local/aperture`. You can change this default directory by editing `Makefile.inc` before building it.

Then run “`make build`” as root to install it. To enable it, add the following line to `/etc/lkm.conf`:

```
/usr/local/aperture/lkm/xf86.o - - /usr/local/aperture/lkm/xf86_mod_install - -
```

and set “`lkm=YES`” in `/etc/rc.conf`

Reboot your system. XFree86 will auto-detect the aperture driver if available.

Warning 1: if you boot another kernel than `/netbsd`, loadable kernel modules can crash your system. Always boot in single user mode when you want to run another kernel.

Warning 2: the aperture driver only allows one access at a time (so that the system is in the same security state once X is launched). This means that if you run multiple servers on multiples VT, only the first one will have linear memory access. Use “`option INSECURE`” if you need more than one X server at a time.

Starting with XFree86 3.9.17, the XFree86 aperture driver also supports MTRR write combining on Pentiums II and AMD K6 class processors.

7.3 MIT-SHM

NetBSD 1.0 and later supports System V shared memory. If XFree86 detects this support in your kernel, it will support the MIT-SHM extension.

To add support for system V shared memory to your kernel add the lines:

```
# System V-like IPC
options      SYSVMSG
options      SYSVSEM
options      SYSVSHM
```

to your kernel config file.

8. Rebuilding the XFree86 Distribution

You should configure the distribution by editing `xc/config/cf/host.def` before compiling. To compile the sources, invoke “`make World`” in the `xc` directory.

8.1 Aperture driver

To build the XFree86 server with the Aperture driver enabled, you should unpack `apNetBSD.shar` and install it first.

Then edit `xc/config/cf/host.def` and add the line

```
#define HasNetBSDApertureDriver    YES
```

to it.

And finally run `make World` in the `xc` directory.

8.2 Console drivers

XFree86 has a configuration option to select the console drivers to use in `xf86site.def`:

- if you're using `pccons` put:

```
#define XFree86ConsoleDefines -DPCCONS_SUPPORT
```

- if you're using `pcvt` put:

```
#define XFree86ConsoleDefines -DPCVT_SUPPORT
```

If you don't define `XFree86ConsoleDefines` in `xf86site.def` the `pccons` and `pcvt` drivers will be supported by default.

8.3 Building on other architectures

Note that the NetBSD project has now its own source tree, based on the XFree86 source tree, with some local modifications. You may want to start with this tree to rebuild from sources. The NetBSD `xsrc` source tree is available at: <ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-current/xsrc/>

9. Building New X Clients

The easiest way to build a new client (X application) is to use `xmkmf` if an `Imakefile` is included in the sources. Type `"xmkmf -a"` to create the Makefiles, check the configuration if necessary and type `"make"`. Whenever you install additional man pages you should update `what.is.db` by running `"makewhat.is /usr/X11R6/man"`.

When porting clients to *BSD systems, make use of the symbol `BSD` for code which is truly BSD-specific. The value of the symbol can be used to distinguish different BSD releases. For example, code specific to the Net-2 and later releases can use:

```
#if (BSD >= 199103)
```

To ensure that this symbol is correctly defined, include `<sys/param.h>` in the source that requires it. Note that the symbol `CSRG_BASED` is defined for *BSD systems in XFree86 3.1.1 and later. This should be used to protect the inclusion of `<sys/param.h>`.

For code that really is specific to a particular i386 BSD port, use `__FreeBSD__` for FreeBSD, `__NetBSD__` for NetBSD, `__OpenBSD__` for OpenBSD, and `__bsdi__` for BSD/386.

10. Thanks

Many thanks to all people who contributed to make XFree86 work on *BSD, in particular, **David Dawes**, **Pace Willison**, **Amancio Hasty**, **Christoph Robitschko**, **Nate Williams**, **Rod Grimes**,

Jack Velte and **Michael Smith**.

CONTENTS

1. What and Where is XFree86?	1
2. Bug Reports for This Document	1
3. New OS dependent features	1
3.1 New OS dependent features in 4.0	1
3.2 New OS dependent features in 3.9.18	1
3.3 New OS dependent features in 3.9.17	1
4. Installing the Binaries	2
5. Configuring X for Your Hardware	2
5.1 About mouse configuration	2
6. Running X	2
6.1 Starting Xdm, the display manager	2
7. Kernel Support for X	3
7.1 Console drivers	3
7.2 Aperture Driver	4
7.3 MIT-SHM	4
8. Rebuilding the XFree86 Distribution	4
8.1 Aperture driver	5
8.2 Console drivers	5
8.3 Building on other architectures	5
9. Building New X Clients	5
10. Thanks	5

\$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/NetBSD.sgml,v 3.55 2000/03/03 21:28:09 dawes Exp \$