README for XFree86 3.1.1 on NetBSD

Rich Murphey, David Dawes, Marc Wandschneider, Mark Weaver, Matthieu Herrb 19 January 1995

Contents

1 What and Where is XFree86?

XFree86 is a port of X11R6 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. The release is available as source patches against the X Consortium X11R6 code, as well as binary distributions for many architectures.

See the *Copyright Notice* (`COPYRIGHT.html`).

The sources for XFree86 are available by anonymous ftp from:

ftp.XFree86.org:/pub/XFree86/current (`ftp://ftp.XFree86.org/pub/XFree86/current`)

Binaries for NetBSD 1.0 are available from:

ftp.XFree86.org:/pub/XFree86/current/NetBSD
(`ftp://ftp.XFree86.org/pub/XFree86/current/NetBSD`)

ftp.laas.fr:/pub/NetBSD/XFree86-3.1.1 (`ftp://ftp.laas.fr/pub/NetBSD/XFree86-3.1.1`)

ftp.iastate.edu:/pub/netbsd/XFree86-3.1.1 (`ftp://ftp.iastate.edu/pub/netbsd/XFree86-3.1.1`)

ftp.eecs.umich.edu:/BSD/XFree86 (`ftp://ftp.eecs.umich.edu/BSD/XFree86`)

If you're still running NetBSD 0.9, you'll have to build it from the source. If you've done it and want to share your binaries, please let us know.

This release has also been tested on NetBSD-current of January, 19th 1995 with `option "COMPAT_10"` in the kernel config file.

XFree86 now also builds on NetBSD/sparc. See section 8.4 for details.

2 Bug Reports for This Document

Send email to *matthieu@laas.fr* (Matthieu Herrb) or *XFree86@XFree86.org* if you have comments or suggestions about this file and we'll revise it.

3 Installing the Binaries

The binary distribution is composed of a number of tar archives which are the executables, servers, fonts, libraries, include files, man pages, config files, and the server link kit. The full distribution takes over 55MB of disk space.

All archives are now made using the NetBSD package tools. We recommend that you install these tools first. All packages can be installed using `pkg_add`.

Note that many clients were moved from the X11R6 distribution to the "contrib" area. However, they are still distributed with the XFree86 binary distribution.

The contents of the packages are:

REQUIRED:

X311doc
READMEs and XFree86 specific man pages.

X311bin
all of the executable X client applications and shared libs

X311font
the misc and 75dpi fonts

X311lib
data files needed at runtime

X311cfg
customizable xinit, xdm, and fs runtime configuration files.

Choose at least one of the following to match your hardware:

X311svga
the 8-bit pseudo-color X server for Super VGA cards

X311vg16
the 4-bit pseudo-color X server for VGA & SVGA cards.

X311mono
the Monochrome X Server

X3118514
the X server for IBM 8514/A and compatible boards

X311agx
the X server for AGX boards

X311mc8
the X server for ATI Mach8 graphics boards

X311mc32
the X server for ATI Mach32 graphics boards

X311mc64
the X server for ATI Mach64 graphics boards

X311p9k
the X server for P9000 based boards

X311s3
the X server for S3 based boards

X311w32
the X server for et4000w32 based boards

OPTIONAL:

X311fs
: the font server with man pages.

X311prog
: the X11 header files and static libraries needed only for compiling other X applications

X311man
: man pages except XFree86 specific ones in the doc archive

X311xfnt
: the other fonts (100dpi, scaled, Japanese, Chinese, etc)

X311link
: X server reconfiguration kit

If this is your first time, then you can safely install all of the packages. As a minimal install, you'll need doc, bin, fonts lib, config, and one X server.

3.1 Installing the packages with `pkg_add`

1. Become "root".
2. For each package that you want to install type:

   ```
   # pkg_add <pkg>.tgz
   ```

   If you don't have enough room in /usr/X11R6 you can specify another directory by using the `-p` option to `pkg_add`. See *pkg_add(8)* for details. Then make a symbolic link `/usr/X11R6` to this directory.

Continue with item 3 below.

3.2 Installing the packages without `pkg_add`

To unpack and install the archives:

1. Create some destination directory in which you'd like the X distribution to live. `/usr/X11R6` is recommended, but if that partition is full (as it is on my machine), then just create an alternate directory, and sym-link `/usr/X11R6` to this place.

   ie, on my machine, I do the following:

   ```
   % cd /usr/local
   % mkdir X11R6
   % ln -s /usr/local/X11R6 /usr/X11R6
   ```

2. You must be logged in as root to unpack the archives and use a "umask" value of 022. Typing "`umask 022`" sets it. *Caution:* If you do not extract the files as user "root" with the correct umask you may not be able to run XFree86 3.1.1. The X server needs special permissions that are only granted to the root user.

Assuming that the destination directory for XFree86 3.1.1 is `/usr/X11R6`, run the following from wherever all the archive files are (replace `/usr/X11R6` with wherever you wish them to go if that isn't the place):

```
cd /usr/X11R6

tar xzf X311doc.tgz
tar xzf X311bin.tgz
tar xzf X311lib.tgz
tar xzf X311font.tgz
tar xzf X311man.tgz

tar xzf X311link.tgz
tar xzf X311prog.tgz
tar xzf X311xfnt.tgz
tar xzf X311fs.tgz
tar xzf X311cfg.tgz
```

And unpack the server(s) you need:

```
tar xzf X311<server>.tgz
```

3. Set the `XWINHOME` environment variable to the path where you installed XFree86. Under csh type:

   ```
   setenv XWINHOME /usr/X11R6
   ```

   under sh type

   ```
   XWINHOME=/usr/X11R6
   export XWINHOME
   ```

   (Replace `/usr/X11R6` by the appropriate dir if necessary).

   Also, don't forget to include these lines in your `.login` or `.profile` for permanent use.

4. Create a symbolic link in `${XWINHOME}/bin` named "`X`" that points to the server that matches your video card: See the *XF86_** man pages for hardware details. For example, if you have an ET4000 based SVGA card:

   ```
   cd $XWINHOME/bin
   rm -f X
   ln -s XF86_SVGA X
   ```

5. Install support for the shared libraries: add the following lines at the end of `/etc/rc.local`:

   ```
   if [ -d /usr/X11R6/lib ]; then
      echo 'Adding X shared libraries'
      ldconfig /usr/X11R6/lib
   fi
   ```

6. Reboot your machine to take all previous modifications into account.

If X was already installed on your machine and you have customized your xinit or xdm runtime configuration scripts, omit the config archive or unpack it elsewhere and extract only that which you need.

> **NOTE:** you don't need to uncompress the fonts files in `lib/X11/fonts`. If you want to uncompress them anyway, don't forget to run "`mkfontdir`" in each directory after that.

4 Installing Xdm, the display manager

Since xdm is dynamically linked against -lcrypt, there's no issue on export restriction outside US for this binary distribution of xdm: it does not contain the DES encryption code. So it's now included in the bin package.

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

You can start xdm automatically on bootup by disabling the console getty adding the following code to `/etc/rc.local`:

```
if [ -x /usr/X11R6/bin/xdm ]; then
  echo -n ' xdm'; /usr/X11R6/bin/xdm
fi
```

To disable the console getty, change "**on**" to "**off**" in the console entry in `/etc/ttys`:

```
ttyv0  "/usr/libexec/getty Pc" pc      off secure
```

5 Configuring X for Your Hardware

The `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.

If you plan to fine tune the screen size or position on your monitor you'll also need the specs for sync frequencies from your monitor's manual.

If your mouse does not work try using kermit or tip to connect to the mouse serial port and verify that it does indeed generate characters.

If you've been running a previous version of XFree86, you can get the video card and monitor information from your previous `Xconfig` file. Don't forget to update the font and RGB paths.

6 Running X

8mb of memory is a recommended minimum for running X. The server, window manager and an xterm take about 4 Mb of memory themselves. On a 4Mb system that would leave nothing left over for other applications like gcc that expect a few meg free. X will work with 4Mb of memory, but in practice compilation while running X can take 5 or 10 times as long due to constant paging.

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.

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
```

The server supports several console drivers: pccons, pcvt, syscons and codrv. 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 1.0. 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.

Syscons and codrv are not bundled with NetBSD 1.0. They are available by anonymous FTP from a number of sites.

If using pccons you get the message:

```
> XFree86 Version 3.1.1 / X Window System
> (protocol Version 11, revision 0, vendor release 6000)
> Operating System: NetBSD
> Configured drivers:
>   Mach8: accelerated server for ATI Mach-8 graphics adaptors
> (Patchlevel 0)
>
> Fatal server error:
> xf86OpenConsole: CONSOLE_X_MODE_OFF failed (Inappropriate ioctl for device)
> Was expecting pccons driver with X support
```

```
> Check your kernel's console driver configuration and /dev entries
>
```

then delete `/dev/ttyv0`. I don't know why "`MAKEDEV`" move `/dev/ttyqf` to `/dev/ttyv0`. (from Hung-Chi Chu *hcchu@r350.ee.ntu.edu.tw*)

7.1 Aperture Driver

By default NetBSD 0.9C and higher include a 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.

The P9000 and AGX servers require linear memory access, other accelerated servers can take advantage of it, but do not require it.

There are two ways to allow XFree86 to access linear memory:

1. Disable the kernel security feature by initializing the "**securelevel**" variable to **-1** in `/sys/kern/kern_sysctl.c`, line 205 and building a new kernel. For more informations, see the comments in `/usr/include/sys/systm.h`.
2. Install the NetBSD aperture driver:
   (a) add the following lines to the end of `/etc/rc.local`:
   ```
   KERNDIR=/usr/X11R6/lib/X11/kernel
   if [ -f ${KERNDIR}/ap.o ]; then
     modload -o ${KERNDIR}/ap -e ap -p ${KERNDIR}/apinstall ${KERNDIR}/ap.o
   fi
   ```
   (b) On pre 1.0 versions of NetBSD modload does not execute post-install scripts. If you're running NetBSD 1.0 or later skip this. Else create the device manually:
   WARNING: be sure to understand what you'll do before proceeding
      i. find out what major device number will be allocated to you by modload. Modload allocates major numbers beginning at **29**. So if "`ap`" is your only device driver module, it will have major number 29. If it's the third, it will be 31...
      ii. goto the `/dev` directory and type: "`mknod xf86 c 29 0`" (replace 29 by the appropriate value if you load more than one device driver module.
   (c) Reboot your system. XFree86 will auto-detect the aperture driver if available.

Caveat: the NetBSD 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.

7.2 MIT-SHM

NetBSD 1.0 supports System V shared memory. If XFree86 3.1.1 detects this support in your kernel, it will support the MIT-SHM extension.

Under NetBSD 1.0, use the following to add shared memory support to your kernel:

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. Then from `/sys/arch/i386/config`, type:

```
# rm -f ../compile/<KERNEL-NAME>/*
# config <KERNEL-NAME>
# cd ../compile/<KERNEL-NAME>
# make depend
# make
```

Then install your new kernel and re-boot:

```
# cp /netbsd /onetbsd
# cp netbsd /
# reboot
```

8 Rebuilding the XFree86 Distribution

The server link kit allow you to rebuild just the X server with a minimum amount of disk space. Just unpack it, make the appropriate changes to the `site.def`, type "`./mkmf`" and "`make`" to link the server. See `/usr/X11R6/lib/Server/README` for more info.

See *INSTALL* (`INSTALL.html`) for instructions on unbundling and building the source distribution.

You should configure the distribution by editing `xc/config/cf/xf86site.def` before compiling. To compile the sources, invoke "`make World`" in the xc directory.

8.1 Console drivers

XFree86 3.1.0C has now 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're using syscons put:

```
#define XFree86ConsoleDefines -DSYSCONS_SUPPORT
```

- if you're running codrv put:

```
#define XFree86ConsoleDefines -DCODRV_SUPPORT
```

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

8.2 console.h and ioctl_pc.h files:

If you want to build a server supporting syscons or codrv and you don't already have the corresponding header files `console.h` and `ioctl_pc.h` installed in `/usr/include/machine`, then install the copies that are supplied in `xc/programs/Xserver/hw/xfree86/etc`. If you run NetBSD-current you probably want to install them in `/usr/src/sys/arch/i386/include` too, so that they get reinstalled each time you make install in `/usr/src/include`.

Another way to keep these files accessible is to make a symbolic link like this in the `xc` directory (suggested by Andrew Cagney *cagney@highland.com.au*):

```
ln -s programs/Xserver/hw/xfree86/etc machine
```

If you have installed the syscons or codrv console drivers, these files should be taken from your installed version of the driver.

8.3 Support for shared libs under NetBSD 1.0

By default XFree86 3.1.1 builds for NetBSD 1.0 with shared libraries support. If you're building on 0.9 or don't want shared libraries add the following line to `xf86site.def`:

```
#define BuildBsdSharedLibs NO
```

8.4 Building on NetBSD/sparc

XFree86 3.1.1 now also compiles on NetBSD/sparc. The Sun server patches from Dennis Ferguson and Matthew Green have been integrated in `xc/programs/Xserver/hw/sun`. Small modifications to `xf86site.def` are needed:

- Set all variables defining the servers to build to **NO**. (The variables controlling the Sun servers to build **Xsun24Server**, **XsunServer** and **XsunMonoServer** are defined at the end of `NetBSD.cf`.)

- Set **ServerToInstall** to the sun server of your choice. (Xsun or XsunMono).
- Look at other applicable options in the *INSTALL document* (`INSTALL.html`).

Problems with this port should be reported to the *port-sparc@NetBSD.Org* mailing list or directly to me *matthieu@laas.fr* rather than to the xfree86 mailing list.

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 `whatis.db` by running "`makewhatis /usr/X11R6/man`".

To avoid the "Virtual memory exhausted" message from cc while compiling, increase the data and stack size limits (in csh type "`limit datasize 32M`" and "`limit stacksize 16M`").

Note: Starting with XFree86 2.1 and NetBSD 0.9A, the symbol **__386BSD__** no longer gets defined either by the compiler or via the X config files for *BSD systems. 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, **__386BSD__** for 386BSD, and **__bsdi__** for BSD/386.

Another note: If you get the message:

```
ld.so: undefined symbol _XtCvtStringToFont
```

at run-time, you've stumbled on a semantic weakness of NetBSD 1.0 dynamic linker. Applications that use libXmu also need libXt. If the client uses a standard `Imakefile`, this dependency will probably by included in the Makefile automagically – you'll not see the problem. Otherwise, just add "`-lXt`" to your library list in the Imakefile or Makefile and relink.

10 Thanks

Many thanks to:

- **Pace Willison** for providing the initial port to 386BSD.
- **Amancio Hasty** for fixing cursor restoration, mouse bugs and many others.

- **Christoph Robitschko** for fixing `com.c` and thus select().
- **Nate Williams** for the patchkit support for X.
- **Rod Grimes** and **Jack Velte** of Walnut Creek Cdrom for use of their machines in preparing the FreeBSD binary release.

`$XConsortium: NetBSD.sgml,v 1.4 95/01/27 16:14:26 kaleb Exp $`

`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/NetBSD.sgml,v 3.10 1995/01/28 16:01:56 dawes Exp $`