# README for XFree86 3.1.1 on FreeBSD 2.0

Rich Murphey, David Dawes 20 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 XFree86 on FreeBSD are available from:

*ftp.XFree86.org:/pub/XFree86/current/binaries/FreeBSD-2.0*
`(ftp://ftp.XFree86.org/pub/XFree86/current/binaries/FreeBSD-2.0)`

*XFree86.cdrom.com:/pub/XFree86/current/binaries/FreeBSD-2.0*
`(ftp://XFree86.cdrom.com/pub/XFree86/current/binaries/FreeBSD-2.0)`

Send email to *Rich-Murphey@Rice.edu* or *XFree86@XFree86.org* if you have comments or suggestions about this file and we'll revise it.

# 2 Installing the Binaries

If you want to save space, first select which archives you want to unpack. If you can't decide what to pick and you have 52Mb of disk space, it's safe to unpack everything.

At a minimum you need to unpack the 'required' `X311*.tgz` archives plus at least one server that matches your vga card. You'll need 9Mb for the minimum required run-time binaries only.

**Required (6.7Mb):**

**X311bin.tgz**
all the executable X client applications and shared libs

**X311fnts.tgz**
the misc and 75 dpi fonts

**X311lib.tgz**
data files needed at runtime

**Required unless you have already customized your configuration files:**

**X311xicf.tgz**
customizable xinit runtime configuration file

**X311xdcf.tgz**
customizable xdm runtime configuration file

**Choose at least one server ( 2.3Mb):**

**X3118514.tgz**
8-bit color for IBM 8514 and true compatibles.

**X311AGX.tgz**
8-bit color for AGX boards.

**X311Mch3.tgz**
8 and 16-bit color for ATI Mach32 boards.

**X311Mch8.tgz**
8-bit color for ATI Mach8 boards.

**X311Mono.tgz**
: 1-bit monochrome for VGA, Super-VGA, Hercules, and others.

**X311P9K.tgz**
: 8, 16, and 24-bit color for Weitek P9000 boards (Diamond Viper).

**X311S3.tgz**
: 8, 16 and 24-bit color for S3 boards (#9 GXE, Actix GE32, SPEA Mercury, STB Pegasus)

**X311SVGA.tgz**
: 8-bit color for Super-VGA cards.

**X311VG16.tgz**
: 4-bit color for VGA and Super-VGA cards

**X311W32.tgz**
: 8-bit Color for ET4000/W32, /W32i and /W32p cards.

**X311nest.tgz**
: A nested server running as a client window on another display.

**Optional:**

**X311doc.tgz**
: (.5Mb) READMEs and XFree86 specific man pages

**X311man.tgz**
: (1.7Mb) man pages except XFree86 specific ones in etc archive

**X311f100.tgz**
: (1.8Mb) 100dpi fonts

**X311fscl.tgz**
: (1.6Mb) Speedo and Type1 fonts

**X311fnon.tgz**
: (3.3Mb) Japanese, Chinese and other non-english fonts

**X311fsrv.tgz**
: (.3Mb) the font server and it's man page

**X311prog.tgz**
: (3.9Mb) config, lib*.a and *.h files needed only for compiling

**X311link.tgz**
: (7.8Mb) X server reconfiguration kit

**X311pex.tgz**
: (.5Mb) PEX fonts and shared libs needed by PEX applications.

**X311lbx.tgz**
: (.2Mb) low bandwidth X proxy server and libraries.

Note that there is no longer a separate xdm archive. FreeBSD 2.0 handles this in shared libraries now, so that the xdm binary does not itself contain des and there is no more need for us to provide separate tar balls.

## 2.1 Full Install:

1. You must be logged in as root to unpack the archives because several executables are set-user-id. Otherwise the server may abort if you unpack it as an ordinary user. You must also use a "`umask`" value of `022` because the X server requires special permissions.

   ```
   % su
   # umask 022
   ```

2. If you have 52Mb free in the `/usr` partition "`cd /usr`" and skip to no. 3. Otherwise, create a directory on another partition and sym link it into `/usr`:

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

3. Unpack everything:

   If you are using sh (as root usually does):

   ```
   # for i in X311*.tgz; do
   #   tar -xzf $i
   # done
   ```

   Else, if you are using csh:

   ```
   % foreach i (X311*.tgz)
   %   tar -xzf $i
   % end
   ```

4. Create a symbolic link "`X`" that points to the server that matches your video card. The *XF86_** man pages list which vga chip sets are supported by each server. For example, if you have an ET4000 based card you will use the XF86_SVGA server:

   ```
   # cd /usr/X11R6/bin; rm X; ln -s XF86_SVGA X
   ```

## 2.2 Minimal Install:

First do numbers 1 and 2 above. Then unpack the required archives:

```
# for i in bin fnts lib xicf; do
#   tar -xzf  X311$i.tgz
# done
```

Then unpack a server archive corresponding to your vga card. The server man pages, `X11R6/man/man1/XF86_*`, list the vga chip sets supported by each server. For example, if you have an ET4000 based card you will use the XF86_SVGA server:

```
# tar -xzf X311SVGA.tgz
# cd /usr/X11R6/bin; rm X; ln -s XF86_SVGA X
```

## 2.3 After either Full or Minimal Install above:

Add /usr/X11R6/bin to the default path for sh in `/etc/profile` and for csh in `/etc/csh.login` if they are not already there:

```
# echo 'set path = ($path /usr/X11R6/bin)' >>/etc/csh.login
# echo 'PATH=$PATH:/usr/X11R6/bin' >>/etc/profile
```

Or make sure all who use X put `/usr/X11R6/bin` in their shell's "path" variable.

Next either reboot or invoke `ldconfig` as root to put the shared libraries in `ld.so`'s cache:

```
# ldconfig /usr/lib /usr/local/lib /usr/X11R6/lib
```

If you had already configured `X11R6/lib/X11/xinit/xinitrc` or `X11R6/lib/X11/xdm/*` omit the `xinit-config` or `xdm-config` archive or unpack it separately and merge in your customizations.

The `fscl` and `f100` archives are optional and can be omitted if you are short on space. The optional `link` archive allows you to reconfigure and customize a X server binary. The optional `prog` archive is needed only for writing or compiling X applications. The optional `pex` archive contains pex clients and libraries for building 3D graphics applications.

> **NOTE:** You don't need to uncompress the font files, but if you uncompress them anyway you must run `mkfontdir` in the corresponding font directory; otherwise your server will abort with the message "`could not open default font 'fixed'`".

If you are running a stock FreeBSD 1.1.5 system you can ignore the rest of this section. If not, make sure you have a spare virtual console which is running a login process (getty). First check how many virtual consoles you have:

```
% dmesg|grep sc
sc0 at 0x60-0x6f irq 1 on motherboard
sc0: VGA color <8 virtual consoles>
```

Then check `/etc/ttys` to make sure there is at least one `ttyv?` devices which doesn't have a getty enabled. FreeBSD 1.1.5 has the first three "on" and the last "off" so that getty runs on only three:

```
% grep ttyv /etc/ttys
ttyv0  "/usr/libexec/getty Pc" cons25  on secure
ttyv1  "/usr/libexec/getty Pc" cons25  on secure
ttyv2  "/usr/libexec/getty Pc" cons25  on secure
ttyv3  "/usr/libexec/getty Pc" cons25  off secure
```

If your kernel has least 4 VTs and only 3 gettys, this is fine. If not, either disable a getty in `/etc/ttys` by changing "on" to "off" or build another kernel with more VTs as described below.

# 3 Installing The Display Manager (xdm)

The display manager makes your PC look like an X terminal. That is, it presents you with a login screen that runs under X.

The easiest way to automatically start the display manager on boot is to add a line in `/etc/ttys` to start it on one of the unoccupied virtual terminals:

```
ttyv4  "/usr/X11R6/bin/xdm -nodaemon" xterm    on secure
```

You should also make sure that `/usr/X11R6/bin/X` is a symbolic link to the Xserver that matches your video card or edit the file Xservers in `/usr/X11R6/lib/X11/xdm` to specify the pathname of the X server.

The change to `/etc/ttys` won't take effect until you either reboot or "`kill -HUP 1`" to force initd to reread `/etc/ttys`. You can also test the display manager manually by loging in as root on the console and typing "`xdm -nodaemon`".

# 4 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.

XFree86 3.1 uses a new configuration file format. Consult the *XF86Config* man page and the general *INSTALL* (`INSTALL.html`) file for instructions.

If you have a `Xconfig` file for XFree86 2.x, use reconfig to translate part of it into the new format:

```
# reconfig <Xconfig >XF86Config
```

and complete the rest according to the *XF86Config* man page and the `XF86Config.sample` file as a template.

In order to protect your hardware from damage, the server no longer will read `XF86Config` files from a user's home directory, but requires that it be in `/etc/XF86Config`, `/usr/X11R6/lib/X11/XF86Config.`*hostname* or `/usr/X11R6/lib/X11/XF86Config`.

You'll need info on your hardware:

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

The easiest way to find which device your mouse is plugged into is to use "`cat`" or "`kermit`" to look at the output of the mouse. Connect to it and just make sure that it generates output when the mouse is moved or clicked:

```
% cat < /dev/tty00
```

If you can't find the right mouse device then use "`dmesg|grep sio`" to get a list of devices that were detected upon booting:

```
% dmesg|grep sio
sio0 at 0x3f8-0x3ff irq 4 on isa
```

Then double check the `/dev` entries corresponding to these devices. Use the script `/dev/MAKEDEV` to create entries if they don't already exist:

```
% cd /dev
% sh MAKEDEV tty00
```

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

# 5 Running X

8mb of memory is a recommended minimum for running X. The server, window manager, display manager and an xterm take about 8Mb of virtual memory themselves. Even if their resident set size is smaller, on a 8Mb system that leaves very space for other applications such as gcc that expect a few meg free. The R6 X servers may 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.

# 6 Rebuilding Kernels for X

The GENERIC FreeBSD 2.0 kernel supports XFree86 without any modifications required. You do not need to make any changes to the GENERIC kernel or any kernel configuration which is a superset.

For a general description of BSD kernel configuration get *smm.02.config.ps.Z* (`ftp://gatekeeper.dec.com/pub/BSD/manuals/smm.02.config.ps.Z`). It is a ready-to-print postscript copy of the kernel configuration chapter from the system maintainers manual.

If you do decide to reduce your kernel configuration file, do not remove the two lines below (in `/sys/arch/i386/conf`). They are both required for X support:

```
options         XSERVER         #Xserver
options         UCONSOLE        #X Console support
```

The generic FreeBSD 2.0 kernel is configured by default with the syscons driver. To configure your kernel similarly it should have a line like this in `/usr/src/sys/i386/conf/GENERIC`:

```
device          sc0     at isa? port "IO_KBD" tty irq 1 vector scintr
```

The number of virtual consoles can be set using the NCONS option:

```
options                 "NCONS=4"               #4 virtual consoles
```

Otherwise, the default without a line like this is 12. You must have more VTs than gettys as described in the end of section 3, and 4 is a reasonable minimum.

The server supports several console drivers: pccons, syscons and pcvt. The syscons driver is the default in FreeBSD 1.1.5 and higher. They are detected at runtime and no configuration of the server itself is required.

The pcvt console driver is bundled in `/usr/ports/util/pcvt` in FreeBSD versions 1.0.2 and above. It can also be ftp-ed from:

*FreeBSD.cdrom.com:/pub/FreeBSD/FreeBSD-1.1/ports/util/pcvt*
`(ftp://FreeBSD.cdrom.com/pub/FreeBSD/FreeBSD-1.1/ports/util/pcvt)`

Refer to the *README.FreeBSD* `(ftp://FreeBSD.cdrom.com/pub/FreeBSD/FreeBSD-1.1/ports/util/pcvt/README.FreeBSD)` file there for complete installation instructions.

The XFree86 servers include support for the MIT-SHM extension. The GENERIC kernel does not support this, so if you want to make use of this, you will need a kernel configured with SYSV shared memory support. To do this, add the following line to your kernel config file:

```
options                 SYSVSHM                 # System V shared memory
options                 SYSVSEM                 # System V semaphores
options                 SYSVMSG                 # System V message queues
```

If you are using a SoundBlaster 16 on IRQ 2 (9), then you need a patch for sb16_dsp.c. Otherwise a kernel configured with the SoundBlaster driver will claim interrupt 9 doesn't exist and X server will lock up.

S3 cards and serial port COM 4 cannot be installed together on a system because the I/O port addresses overlap.

# 7 Rebuilding XFree86

The server link kit allows you to build an X server using a minimum amount of disk space. Just unpack it, make the appropriate changes to `site.def`, type "`./mkmf`' and "`make`" to link the server. See *README.LinkKit* `(LinkKit.html)` for more info.

The source tree takes about 114Mb before compiling and an additional 100Mb after "`make World`". You should configure the distribution by editing `xf86site.def` and `site.def` in `xc/config/cf` before compiling. By default, the config files are set up to build shared libraries. If you are running a version of FreeBSD that doesn't include shared library support, add the following line to `site.def`:

```
#define BuildBsdSharedLibs          NO
```

If your system doesn't have support or SYSV shared memory (for example, if you don't have the `<sys/shm.h>` header), you should disable the MIT-SHM extension by adding the following line to `site.def`:

```
#define HasShm                      NO
```

To compile the sources on FreeBSD 1.1 and later, type:

```
make World
```

If you are running an old version of FreeBSD (before 1.1), then type:

```
make World BOOTSTRAPCFLAGS=-D__FreeBSD__
```

# 8 Building Other X Clients

The easiest way to build a new client (X application) is to use `xmkmf` if an `Imakefile` is included with it. Type "`xmkmf -a`" to create the Makefiles, then type "`make`". Whenever you install additional man pages you should update `whatis.db` by running "`makewhatis /usr/X11R6/man`".

On FreeBSD 1.0 and earlier systems, 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`"). This is not needed on FreeBSD 2.0 and later since the defaults are ample.

**Note:** Starting with XFree86 2.1 and FreeBSD 1.1, the symbol **__386BSD__** no longer gets defined either by the compiler or via the X config files for FreeBSD 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.

# 9 Thanks

Many thanks to:

- **Pace Willison** for providing initial *BSD support.

- **Amancio Hasty** for 386BSD kernel and S3 chipset support.
- **David Greenman, Nate Williams, Jordan Hubbard** for FreeBSD kernel support.
- **Rod Grimes**, **Jordan Hubbard** and **Jack Velte** for the use of Walnut Creek Cdrom's hardware.
- **Orest Zborowski**, **Simon Cooper** and **Dirk Hohndel** for ideas from the Linux distribution.

`$XConsortium: FreeBSD.sgml,v 1.3 95/01/23 15:34:41 kaleb Exp $`

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