

X Window System Installation

Release 3

Version 10

Jim Gettys

Digital Equipment Corporation
MIT Project Athena

ABSTRACT

X is a network transparent window system developed at MIT which runs under 4.3BSD UNIX® and Ultrix-32‡ 1.2. This document details the installation under 4.3BSD. This document also presumes Vs100 hardware.

1. Overview

Given an X distribution hierarchy, the following basic steps need to be performed:

- 1) decide where you want to install fonts (and firmware if you are using VS100 displays).
- 2) configure the site file(s) for the displays you use for this information.
- 3) build all software.
- 4) install the software into the directories.
- 5) add a device driver or configure the device driver into your kernel.
- 6) reboot your machine.
- 7) rename pseudo-teletypes for your login window.
- 8) test that everything is working.
- 9) configure your /etc/ttys file for the display.

By default, X is distributed to install on a stock 4.3BSD system; installation under Ultrix-32 release 1.2 should essentially be the same.

2. Overview of the Software Hierarchy.

The device independent part of the X server is contained in the **X/X** directory. Device dependent code for different displays are contained in directories **libvs100**, **libqvss**, **libqdss**, **libsun** and eventually others. The device independent code is linked against these libraries to form executable programs for each display type; for example **Xvs100** or **Xqvss** or **Xsun**. In each of these libraries is an include file named **vssite.h** which has definitions of the directory where fonts (and if any firmware is needed, where the firmware file) are located. These files should be edited to suit local taste. By default, as distributed, fonts

‡ Ultrix and Ultrix-32 are trademarks of Digital Equipment Corporation
Copyright © 1985, Massachusetts Institute of Technology

Permission to use, copy, modify and distribute this documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of the software described herein for any purpose. It is provided "as is" without express or implied warranty.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.

are located in the directory `/usr/new/lib/X/font`.

The directory **Xlib** contains C client library code for client programs. As distributed, all code links against the 'local' copy of the library, and does not require that the library be installed in `/usr/lib` for linking.

The directory **uwm** contains the "Ultrix Window Manager", hands down the best window manager written to date.

The directory **XMenu** contains a deck of cards menu facility.

The **cursor** directory contains cursor bitmaps used by various of the client programs.

The directory **font** contain fonts in the appropriate format for the Vs100 display.

The directory **inline** contains a slightly modified version of the 'inline' program used to make inline code optimizations in the 4.3BSD kernel. This program is used by several libraries to avoid procedure call overhead in critical code of the window system. As this version is known to have some bugs in it, we do not recommend using it elsewhere.

The **rgb** directory contains a color database, to provide translation of english names to RGB colors. This is used by the X server to support color name lookup for client programs. It is a general database, and not directly tied to X per se'.

The directory **s-code** contains the firmware file for the DEC Vs100 display.

The **src** directory contains modified 4.2BSD code, which you may want to install (xcomsat, for example).

The **sys** directory hierarchy contains system files in their normal arrangement; for example, the Vs100 device driver, an improved 4.2BSD pseudo teletype driver, and so on. Only files which do not exist on the 4.3BSD distribution are included, as most files required have already been installed.

All of the directories starting with lower case "x" contain various client programs for X; two different window manager programs, terminal emulator, performance monitors, window dump and undump programs and the like. Xted is a version of the Clu screen editor "ted" which has been modified to interact directly with the window system.

The `exe.v10` directory contains executables for the Clu programs, since most people do not have CLU compilers.

The directory **CLUlib** contains CLU client library code. Clu is available from MIT under license. On the distribution tape, only two demo programs (**xdemo** and **xfax**), and a screen editor (**xted**) are provided in their directories written in Clu. The facsimile files used by the **xfax** demo are in the **fax** directory. All "serious" code is written in the obsolete (but widely available) C language.

3. Installation Steps

This chapter explains in greater detail the steps covered in the overview. It presumes the source hierarchy has been loaded in some convenient location, and that the `/usr/new` directory and `/usr/new/lib` directory already exist. As distributed, the X makefile only creates `Xvs100`. If you have source or object for other display types in their device dependent directories, you should edit the `X/X/Makefile` to build them.

3.1. Prelude

NOTE: As X is a network transparent window system, client programs use network facilities to communicate with the window system. Make sure your network code works properly BEFORE attempting to use X. For example, `talk(1)` should be working properly between normal terminals, or you should be able to `rlogin` to either the local or to another machine if you have a network.

Also make sure you have made as many pseudo-teletypes as possible (`cd /dev; MAKEDEV pty0; MAKEDEV pty1`). These are used for terminal emulator windows, and it is possible you may use quite a few of them.

If you have other machines in your network that run a 4.2BSD derived system, you can also move the C language source code and recompile it on those machines. The C applications should be able to

communicate with the X server on your display. This has only currently be tested for Sun and IBM RT/PC's, but in principle, you should be able to interoperate properly with your Vax, Sun, or other displays.

3.2. Font and Firmware Files

For each device type supported by X, fonts may differ. There may also be local installation restrictions which may prohibit you from using the default location of `/usr/new/lib/X` for the fonts or other files needed by X (for example, Vs100 firmware). In each device dependent X library should be a file `vssite.h` which can be tailored to find the fonts and firmware in a different location. Another reason why you may want to change the font and firmware directory locations would be to allow use of the display while not having all of your file systems mounted. Tailor this file to your taste, and modify the master make file to move the fonts and firmware to your location.

The binaries are normally installed into `/usr/new`. If you want to change this, edit the master Makefile and change `CONFDIR` to the directory you choose. The entry "make reconfig" can be used to automatically edit the Makefiles in the directories from `/usr/new` to your new configuration.

3.3. Building Software

To build executable versions of all X programs, execute the command "make all" in the main directory.

Compiling all software takes quite a while. Building the X library takes the longest, as there are more than one hundred fifty modules. This should complete without error.

3.4. Installing the X Executables

As super user, execute the command "make install". This should complete without error.

This also copies all necessary files for users to program using X into `/usr/include/X`.

3.5. Building a Kernel With the Device Driver

You must add a line to your configuration file for each display you have. Make sure the CSR address matches between your configuration file and your hardware. Configure, make, and install the kernel containing the display driver. When you reboot the machine, make sure that your display auto configures during boot.

You should also make a device entry for each display. For a Vs100, change directory to `/dev`, and perform a "MAKEDEV vs?", where '?' is the number of the Vs100 as root. For a QVSS on a MicroVAX, the command would be `"/etc/mknod /dev/mouse c 35 2"`. Normally, the protection on the device would be only user read/write, but for debugging purposes you may want to temporarily change it.

3.6. Renaming Pseudo Teletypes

To avoid a possible race condition, and to allow consistent Unix program behavior, we dedicate a pseudo teletype for each display's login window. All other pty's are allocated dynamically when used. Pseudo TTY's come in pairs, the master and the slave. We rename them to be "ttyv?" where '?' is the number of the display.

So for example, we might execute the commands: `"mv ttyqf ttyv0; mv ptyqf ptyv0"`, and similarly for any other displays. When logged in, you would appear to be logged in on "ttyv0" in this case. We use the last pseudo teletypes since all other utilities start searching from lower to higher, and we'd just as soon have them find a pty as soon as they can.

3.7. Testing and Trouble Shooting

We highly recommend testing before attempting to enable login on the display. Error messages will go to your terminal, rather than being logged in a file. You can use `xinit(1)` to aid you in testing, but it is most easily performed from another terminal or from across the network.

To test a Vs100 on line 0, for example, you would change directory to `/usr/new` (or wherever you decided to put the executable programs), and would type `"Xvs100 0 0 &"`. (For a MicroVAX, the

command would be "Xqvss 0 0 &"). The first argument is which device to use (in this case */dev/vs0* will be used). There can only be one display on a MicroVAX. The second argument is ignored. See the *X(8c)* manual page for other options.

If everything succeeds, you should get a grey background and a large X cursor on the screen. For reasons we have never understood, it may take three tries to get a VS100 display to respond. If not, the most common mistake is the fonts or firmware to be in the wrong location or a directory or file to not be readable.

You should now be able to run any of the X programs. A common beginning mistake would be to forget to set the "DISPLAY" environment variable. Most programs also take arguments to specify the host and display number. So, for convenience while testing, you might execute the command "setenv DISPLAY *yourhost:0*" where *yourhost* is the name of your machine when using the C-shell. This variable will be set for you automatically when you log in in the future.

A common problem that might prevent the *xterm(1)* program from working is it not being set user ID and owned by root. The installation makefiles should be installing *xterm(1)* this way.

If everything is working, you should be ready to enable the line for login.

3.8. Configuring Lines in */etc/ttys*

If you started X in the previous step, you will want to abort it now. For each display you have on a machine, you must have a line in */etc/ttys* to drive the terminal emulator used for login and the window system server. NOTE: The format of the */etc/ttys* file has changed radically between 4.2 and 4.3. You cannot set up a display for login on a 4.2BSD system without installing new versions of */etc/init*, */etc/getty*, and */bin/login* from 4.3.

An example line in */etc/ttys* is given in the *X(8c)* manual page (though you will have to customize the line for the location and names of the executable programs). An example for a Vs100 installed without any reconfiguration on 4.3BSD might be:

```
ttv0 "/usr/new/xterm -L =-1+1 :0" xterms on window="/usr/new/Xvs100 0"
```

You can customize these commands to taste.

You can tell *init(8)* to re-read the */etc/ttys* file by the command "kill -1 1" as super user.

4. Installation Under 4.2BSD

You are essentially on your own here. Besides the steps outlined in sections one through three, you must install the Vs100 device driver, install an improved version of the pseudo-tty driver, and even then you will run into an obscure TCP problem that will cause X to be unreliable.

We have also decided that it is hazardous for most people to attempt installation of the changes to *init*, *getty* and *login* required for automatic login on the display. Furthermore, due to local changes and version skew, we no longer have a consistent set of sources for these programs for 4.2, and have no good way to test any installation directions as we have been running these changes for nearly a year on all our machines. For these reasons, WE RECOMMEND UPGRADING to 4.3BSD or Ultrix 1.2 where these changes are standard.

5. Installation Under Ultrix-32 1.1

The name of the executable built in the "X" directory for QVSS is Xqvss. Note that some of the files mentioned here are only available on the MIT distribution of X.

We recommend upgrading to Ultrix-32 1.2 as soon as possible. When you do, be sure to rebuild *Xqvss* and *xterm*.

If you need to relocate the font directory, edit the file *libqvss/vssite.h*, recompile *libqvss/ddxfontloc*, and replace the file into *libqvss/libqvss.a* using *ar(1)*.

Note that the *Xqvss* server and the *xterm* executable should be recomiled when you upgrade to 1.2. A server built under 1.1 will not execute under field test two of Ultrix-32 1.2 or under version 1.2 release due

to a system call change.

The improved `pty` driver has already been installed, but the TCP problem is still with you. X will be slightly unreliable when pushed hard until you upgrade to Ultrix 1.2 when using the TCP domain on the local host. Everything should run reliably if you run using the Unix domain. (use a `DISPLAY` variable of `unix:0`). The file provided should be added to your `BINARY.mvax` directory, and your kernel rebuilt. This will change the keyboard handling in the shared event queue of the QVSS driver to match what this version of X requires.

In addition, the file `sys/vaxuba/qevent.h` should replace the file of that name in `/usr/include/vaxuba`. This file avoids the redefinition problems you would encounter with the original. The bit-blit code is not available in source. There is an object library `X/libqvss/libqvss.a` containing the object of the device dependent library. Note that you should not just `rcp(1)` the distribution tree, as there are symbolic links in the tree. If you have, you may need to `ranlib(1)` the QVSS object library.

6. Installation Under Sun 2.X and 3.0.

The name of the executable built in the “X” directory for the is Xsun.

Use under `init(8)` is not supported. Use `xinit(1)` instead. See the caveats in the `Xsun(8c)` manual page.

Be sure to rebuild `Xsun` and `xterm` under Sun 3.0.

As the Sun version of X happened at the last possible moment, its installation is not as smooth as one would like.

The top level `X/Makefile` has a “make sun” entry which edits out an include of `<sys/types.h>` which should be executed on Sun 2.X versions of the system. You should remove the build of `inline`, `libvs100`, and `xpr` from the `X/Makefile`. You must also edit out the use of “inline” in `X/Xlib/Makefile`, which is a VAX specific optimizer.

Appendix A - Location of files

This section documents where any files NOT installed somewhere in the `/usr/new` directory tree.

<code>/usr/include/X/*</code>	Include files for X programming.
<code>/usr/lib/rgb.*</code>	Color database.
<code>/usr/lib/Xkeymap.txt</code>	Sample Keymap file.
<code>/usr/lib/libX.a</code>	X library.
<code>/usr/lib/libX_p.a</code>	Profiled X library.
<code>/usr/lib/libXMenu.a</code>	Menu library.
<code>/usr/lib/libXMenu_p.a</code>	Profiled Menu library.
<code>/usr/man/man/...</code>	Manual pages for man(1) pages.
<code>/usr/man/man3/...</code>	Manual pages for libraries.
<code>/usr/man/man8/...</code>	Manual pages for servers.

`Xterm(1)` is the only file installed set user id to root. `Xload` must be set group id to “kmem”.