X Window System Installation
Release 4
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 and many other systems. This document details the installation on systems supported by the MIT distribution.

1. Overview

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

1) read in the tape.
2) decide where you want to install fonts (and firmware if you are using VS100 displays). We recommend leaving this alone unless you have some good reason.
3) configure the site file(s) for the displays you use for this information.
4) build all software.
5) install the software into the directories.

The following steps may not be needed (or possible) on some systems.

6) add a device driver or configure the device driver into your kernel.
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 so on. 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

‡ Ultrix and Ultrix-32 are trademarks of Digital Equipment Corporation

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 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 library. As this version is known to have some bugs in it, we do not recommend using it elsewhere, and the current distribution only uses it on Vaxes.

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 a few modified files.

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. 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.

This distribution was tested under the following versions of different vendors systems.

4.3BSD
Ultrix 1.2
Sun 3.0
Apollo 9.5
Integrated Solutions 3.07
IBM 4.2A Release 2.

You are on your own if you are using versions prior to those listed above. We recommend upgrading to the later releases. In particular, the server will not run on IBM 4.2A release 1, and previous releases of the other systems may or may not work properly. It is impossible for us to keep straight all the different versions of different vendor's systems.

This distribution has not been tested for the Integrated Solutions machine, as we had none available for testing.

The Apollo testing showed the installation to be quite rocky, due to some problems with their C preprocessor, and there was not time for complete integration of the changes required. Good luck; you may need it, but it has been seen to work.

The client code should be easily portable to other 4.2BSD based systems, the CCI for example. Some System V based Unix systems also have the Berkeley network support in them, but you are on your own here.

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. Performance of the terminal emulator will be improved by use of 4.3BSD's improved pty driver, which is six times faster than the 4.2BSD version. The C applications should be able to communicate with the X server on your display.

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. Reading the distribution.

The software comes on a single 1600 BPI magnetic tape in *tar*(1) format. All files will be extracted into a directory named **X**. An example command would be:

- tar xf /dev/rmt8

If you move the distribution to a different machine than the one you read the tape on, use care to preserve the symbolic links, either by moving the distribution as a single file or by using *tar* across the net. If you don't, you will end up with more than one copy of certain key include files, and may get confused later if you make changes.

3.4. Building the server

If you only want to build the client programs on a machine that does not have a display, you can skip this step.

If you have a display on your workstation, you can build a server for your display. The types of displays supported under the MIT distribution include the DEC VS100, VS-1, VS-2 and VS-2 RC, most Sun workstation displays, the Apollo displays, the IBM RT/PC displays under 4.2A only (this distribution will NOT run under AIX!), and the Integrated Solutions Optimim V workstations. Servers for other displays may only be available from the manufacturers. Examples of this include the Vaxstation II/GPX from Digital, the HP Bobcats, and Sony displays.

To build the server for your machine, type one of the following:

- make dec # for Digital Vs100 and VS1 and VS2
- make sun # for Sun workstations
- make apollo # for Apollo Computer workstations
- make is # for Integrated Solutions workstations
- make ibm # for IBM RT/PC under 4.2A

This should complete with no errors on the DEC, Sun, and IBM workstations. The Apollo compiler may generate a number of warnings, and X/dispatch.c takes a LONG time to compile. The Apollo preprocessor has problems with several macros new to release 4; you will have to edit out all occurences of "B16", and "UBPS" should be changed to 1. These macros are defined to make it easier to port X to machines where a short is not 2 bytes.

3.5. 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 on most machines.

3.6. 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*.

You should also install the correct termcap and terminfo (if applicable) files out of *xterm/termcap* and *xterm/terminfo* into */etc/termcap* and your terminfo database if they are not already there.

The *xinit* program, useful on machines without the proper init support for login, expects that the running server to be called "X". You can either rename the appropriate server for your display, or use the correct arguments to *xinit*.

3.7. Building a Kernel With the Device Driver

On some machines, the display driver or other auxilary file may not come configured into the system or other device files may not exist. 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. The VS100 driver comes with 4.3BSD. 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 # if /dev/mouse does not exist on a VS-2.
- /etc/mknod /dev/bell c 12 2 # for bell to ring on a Sun.
- MAKEDEV displays # for displays on the RT/PC

Normally, the protection on the device would be only user read/write, but for debugging purposes you may want to temporarily change it. On a DEC VS-2, you should also adb /sys/vaxuba/qv.o and change the variable "qv_def_scrn" to 2 and rebuild your system. This will cause the correct screen parameters to be used on the VR-260 monitor.

3.8. 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 the file */usr/adm/X?.hosts*. You can use *xinit(1)* to aid you in testing, but it is most easily performed from 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)* and the manual pages for particular servers 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. *Xterm* may also fail if the files described in the previous section are not consistent, or if you attempt to use an xterm built before the */etc/init* changes were installed.

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

3.9. Automatic Login Support

Some systems are capable of starting the server and a login *xterm* automatically, in particular 4.3BSD and Ultrix 1.2 and later. If your system does not support the new /etc/ttys format, you should skip this section. On other systems, if you have source for 4.3BSD you may want to install this support. Mere mortals should probably think long and hard before risking such installation, but wizards may find it not too difficult. Affected programs include */etc/init*, */etc/getty*, */bin/login*, and C library routines *ttyname*(3), *isatty*(3), *ttyslot*(3) and all programs that depend on the format of */etc/ttys*, */etc/gettytab* and */etc/utmp*. The programs need to be installed as a set, and *xterm* must be recompiled.

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. You will use many pty’s, so make as many as possible. 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:

- cd /dev
- MAKEDEV pty0
- MAKEDEV pty1
- mv ttyqf ttyv0 # may already exist on some machines
- 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.

3.10. 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:

```
ttyv0 "/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.

Appendix A - Location of files

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

/usr/include/X/*	Include files for X programming.
/usr/lib/rgb.*	Color database.
/usr/lib/Xkeymap.txt	Sample Keymap file.
/usr/lib/libX.a	X library.
/usr/lib/libX_p.a	Profiled X library.
/usr/lib/libXMenu.a	Menu library.
/usr/lib/libXMenu_p.a	Profiled Menu library.
/usr/man/mann/...	Manual pages for man(1) pages.
/usr/man/man3/...	Manual pages for libraries.
/usr/man/man8/...	Manual pages for servers.
/usr/games/xtrek	If Xtrek is installed.
/usr/games/lib/xtrek	

Xterm(1) is the only file installed set user id to root. *Xload* must be set group id to ‘‘kmem’’.