X Window System, Version 11

Release 6.4

Release Notes

The Open Group X Project Team

30 January, 1998

-- --

1. What Is X11

X11, or X, is a vendor-neutral, system-architecture neutral network-transparent window system and user interface standard. In other words it's windows for UNIX. But X is not just for UNIX — X runs on a wide range of computing and graphics machines including Macintosh, OS/2, Microsoft's MS-Windows family of operating systems, and almost all of the so-called Network Computers. X can use your network — you may run CPU-intensive programs on high powered workstations and display the user interface (the windows) on inexpensive desktop machines.

2. What is Release 6.4

Release 6.4 (R6.4) is The Open Group X Project Team's update to the X Consortium's Release 6.3 (R6.3) of X11 and all prior releases. It is compatible with with all releases going back to R1 at both the source and protocol levels.

The X Consortium was an independent, not-for-profit membership corporation formed in 1993 as the successor to the MIT X Consortium. It was dissolved at the end of 1996 and all assets such as trademarks and copyrights were transferred to The Open Group.

The Open Group's X Project Team was formed to continue maintenance and development of X. Membership in The Open Group X Project Team does not require membership in The Open Group. To join the X Project Team contact the sales office near you — you can find a list of sales offices at http://www.opengroup.org/contacts/sales.htm, or download the membership kit from http://www.opengroup.org/tech/desktop/x/projteam.htm.

Instructions for building and installing R6.4 can be found in the file INSTALL.PS (PostScript) or INSTALL.TXT (plain text), available separately and also contained in the release.

3. Overview of the X Project Team Release

Like all the releases that preceded it, R6.4 is a source code release. In order to use the release it is necessary to first unpack the distribution, compile it, and then install it. The source contains the following items:

X Consortium Standards and X Project Team Specifications

The X Consortium produced standards — documents which define network protocols, programming interfaces, and other aspects of the X environment. These standards continue to exists in the X Project Team release. The X Project Team produces specifications. Like X Consortium Standards, these are documents which define network protocols, programming interfaces, and other aspects of the X environment. Under the aegis of The Open Group, X Consortium standards, X Project Team specifications, and other specifications are the basis for portions of The Open Group's various CAE Specifications. Some of the new features in this release are not *standard*, that is, there is no accompanying specification.

Implementations

For most of X Consortium standards and X Project Team specifications, a high-quality implementations is provided to demonstrate proof of concept, to give vendors a base to use, and early adopters a chance to begin developing using the new features. These are not *reference* implementations — the written specifications are the authoritative definition.

Fonts

A collection of bitmap and outline fonts are included in the distribution, contributed by various individuals and companies.

Utility Libraries

A number of libraries, such as Xmu and the Athena Widget Set, are included. These are not standards, but are used in building the applications contained in the release and may be useful in building other applications.

Programs

We also provide a number of application programs. A few of these programs, such as *xdm* (or its equivalent), should be considered essential in almost all environments. The rest of the applications carry no special status; they are simply programs that have been developed and/or maintained by X Consortium and X Project Team staff. In some cases, you will find better substitutes for these programs contributed by others.

4. Supported Operating Systems

This release was built and tested on the following reference platforms:

Digital Unix 4.0A
Fujitsu UXP V20L10
HPUX 10.20
Solaris 2.5

This release was also built on the following systems:

AIX 4.2
IRIX 6.2
FreeBSD 2.2.2
S.u.S.E. Linux 5.0.0 (kernel 2.0.30, libc 5.4.33)
SunOS 4.1.4
Windows NT 3.51 (NCD WinCenter)

In all cases except SunOS we have used the vendor's compiler. On SunOS we build with the GNU C compiler (*gcc*).

5. Supported Graphics Devices

This release includes the necessary device-dependent support to build a native X server for the following platforms:

HP-UX: Xhp
Digital Unix: Xdec on DECstation 3000/400 (Alpha) with PMAG-B
SunOS/Solaris: Xsun — see the Xsun man page for supported cards
XFree86: See the XF_* man pages for supported cards

In addition to the above, the Xvfb and Xnest servers can be built on all platforms.

Native X servers are not built on AIX, Fujitsu UXP, IRIX, or Microsoft Windows NT.

6. The Source Tree

The source is distributed in UNIX tar files. The source unpacks from the tar files into a source tree, and the name of the base directory of the source tree is **xc**. The name **xc** as the base of the source tree has been retained from the X Consortium releases.

The general layout under **xc/** is as follows:

config/	imake config files, *imake*, *makedepend*, etc.
doc/	all documentation other than per-program manual pages
fonts/	BDF, Speedo, Type1 fonts
include/	common include files
lib/	libraries
nls/	national language support files
programs/	all programs, including the X server and *rgb*,
util/	*patch*, *compress*, other utilities
bug-report	bug reporting template
registry	X Registry

7. X Registry

The X Project Team maintains a registry of certain X-related items to aid in avoiding conflicts and to aid in sharing of such items. The registry is in the file **xc/registry**.

8. Extensions Supported

Release 6.4 includes source for the following extensions: BIG-REQUESTS, DOUBLE-BUFFER, DPMS, Extended-Visual-Information, LBX, MIT-SHM, MIT-SUNDRY-NONSTANDARD, Multi-Buffering, RECORD, SECURITY, SHAPE, SYNC, TOG-CUP, X3D-PEX, XC-APPGROUP, XC-MISC, XFree86-VidModeExtension, XIE (X Image Extension), XINERAMA. XInputExtension, XKEYBOARD, XpExtension (printing), XTEST, and XTestExtension1,

Not all of these extensions are standard; see the Standards manual page. Some of these extensions may not be supported on every platform.

9. Implementation Dependent Parameters

Some of the specifications define some behavior as implementation-dependent. Implementations of the X Consortium standards and X Project Team specifications must document how those parameters are implemented.

The default values in this release of the implementation dependent parameters are:

XFILESEARCHPATH default:

This default can be set at build time by setting the *imake* variables XFileSearchPathDefault, XAppLoadDir, XFileSearchPathBase, and ProjectRoot in **xc/config/cf/site.def**. See

xc/config/cf/README for instructions and **xc/config/cf/X11.tmpl** for details of how these configuration variables are used.

By default the imake variable ProjectRoot is **/usr/X11R6.4** and XFILESEARCHPATH has these components:

$ProjectRoot/lib/X11/%L/%T/%N%C%S
$ProjectRoot/lib/X11/%l/%T/%N%C%S
$ProjectRoot/lib/X11/%T/%N%C%S
$ProjectRoot/lib/X11/%L/%T/%N%S
$ProjectRoot/lib/X11/%l/%T/%N%S
$ProjectRoot/lib/X11/%T/%N%S

XUSERFILESEARCHPATH default:

If the environment variable XAPPLRESDIR is defined, the default value of XUSERFILESEARCHPATH has the following components:

$XAPPLRESDIR/%L/%N%C
$XAPPLRESDIR/%l/%N%C
$XAPPLRESDIR/%N%C
$HOME/%N%C
$XAPPLRESDIR/%L/%N
$XAPPLRESDIR/%l/%N
$XAPPLRESDIR/%N
$HOME/%N

Otherwise it has these components:

$HOME/%L/%N%C
$HOME/%l/%N%C
$HOME/%N%C
$HOME/%L/%N
$HOME/%l/%N
$HOME/%N

XKEYSYMDB default:

Defaults to *$ProjectRoot***/lib/X11/XKeysymDB**.

XCMSDB default:

Defaults to *$ProjectRoot***/lib/X11/Xcms.txt**.

XLOCALEDIR default:

Defaults to the directory *$ProjectRoot***/lib/X11/locale**. The XLOCALEDIR variable can contain multiple colon-separated pathnames.

XErrorDB location

The Xlib error database file is *$ProjectRoot***/lib/X11/XErrorDB**.

XtErrorDB location

The Xt error database file is *$ProjectRoot***/lib/X11/XtErrorDB**.

Supported Locales

Locales supported by this implementation are in **xc/nls/locale.dir**. The mapping between various system locale names and X locale names is in **xc/nls/locale.alias**. Both files are installed in the default XLOCALEDIR directory, i.e. *$ProjectRoot***/lib/X11/locale/**).

Supported Input Methods

This distribution does not include source for any input method servers; however Xlib supplies a default built-in input method that supports compose processing in 8-bit locales. Compose files are provided for Latin-1 and Latin-2. The built-in input method can support other locales, given suitable compose files. See **xc/nls/Compose/iso8859-*** for the supported compositions.

The Input Method Server Development Kit (IMdkit) is at ftp://ftp.x.org/pub/unsupported/lib/IMdkit/.

10. What Is New in Release 6.4

This section describes changes in the X Project Team distribution since Release 6.3.

The major new functionality in R6.4 is: Display Power Management Signaling (DPMS) to set "green" computer monitors into power saving mode; Extended Visual Information to allow applications to discover more about the graphics capabilities of the server than the core protocol allows; Colormap Utilization Policy (TOG-CUP) allows applications to discover desktop special colors, e.g. MS-Windows reserved (pre-allocated) colors on PC-Xservers, and store read-only (sharable) colors in specific locations in a colormap; and Xinerama, a wide screen server that combines two or more screens into a single virtual screen.

The X Toolkit Intrinsics library (libXt) now has IBM's Easy Resource Configuration support included.

Xlib (libX11) has two new APIs: XkbSetPerClientControls and XkbGetPerClientControls. These two functions were unintentionally omitted from the library in previous releases.

The XFree86 servers are now based on XFree86 3.3.1.

11. What is Unchanged in Release 6.4

As this is an update release, there is a great deal of stability in the standards, libraries, and clients. No existing standards have changed in a material way; although some documents have been updated with minor corrections. The extension library, *libXext*, is updated to include the DPMS, Extended-Visual-Information, TOG-CUP, and XINERAMA extension interfaces. All previous interfaces in these and all other libraries are unchanged.

12. New OS Support

The following table shows the versions of the operating systems that were used to develop this and prior releases:

System	R6	R6.1	R6.[23]	R6.4
AIX	3.2.5	4.1.4	4.2	4.2
A/UX	3.0.1	–	–	–
BSD/386	1.0	–	–	–
Digital Unix (OSF/1)	1.0/1.3	3.2C	4.0A	4.0A
FreeBSD	–	2.1.0	2.1.6	2.2.2
Fujitsu UXP	–	–	–	V20L10
HP-UX	9.1	10.01	10.01	10.20
IRIX	5.2	5.3	6.2	6.2
Linux (kernel)				
Slackware 2.3	–	1.2.11	–	–
Slackware 3.1	–	–	2.0	–
S.u.S.E. 5.0	–	–	–	2.0.30
Mach	2.5	–	–	–
NEWS-OS	6.0	–	–	–
Solaris	2.3	2.4	2.5	2.5
SunOS	4.1.3	4.1.3	4.1.4	4.1.4
Ultrix-32	4.3	4.4	–	–
UNICOS	8.0	–	–	–
Unixware SVR4.2	1.0	2.02	2.02	–
Windows NT	3.1	3.5	4.0	3.51

13. New Specifications

The following are the new X Project Team specifications in Release 6.4. Each is described in its own section below.

Display Power Management Signalling (DPMS)
Extended Visual Information (EVI)
Colormap Utilization Policy (TOG-CUP)

13.1. Display Power Management Signaling

This extension provides X Protocol control over the VESA Display Power Management Signaling (DPMS) characteristics of video boards under control of the X Window System.

Traditionally, the X Window System has provided for both blanking and non-blanking screen savers. Timeouts associated with these built-in screen saver mechanisms are limited to idle (dwell) time, and a change timeout that specifies the change interval for non-blanking screen savers.

The United States' Environmental Protection Agency (EPA) Energy Star program requires that monitors power down after some idle time by default. While it is possible to simply overload the existing screen saver timeouts, this solution leaves the non-privileged user little to no control over the DPMS characteristics of his or her system. For example, disabling DPMS would require some unintended side effect in the core screen saver, such as disabling the changing of a non-blanking screen saver.

13.2. Extended Visual Information

The Extended Visual Information (EVI) extension allows a client to determine information about core X visuals beyond what the core protocol provides.

As the X Window System has evolved, it has become clear that the information returned by the core X protocol regarding Visuals is often insufficient for a client to determine which is the most appropriate visual for its needs. This extension allows clients to query the X server for additional visual information, specifically as regards colormaps and framebuffer levels.

This extension is meant to address the needs of pure X clients only. It is specifically and purposefully not designed to address the needs of X extensions. Extensions that have an impact on visual information should provide their own mechanisms for delivering that information. For example, the Double Buffering Extension (DBE) provides its own mechanism for determining which visuals support double-buffering.

13.3. Colormap Utilization Policy

This extension has three purposes: a) to provide mechanism for a special application (a colormap manager) to discover any special colormap requirements, e.g. the colormap entries that are nominally reserved for desktop colors in the MS-Windows environment and initialize the default colormap so that it can be more easily shared; and b) to encourage colormap sharing and reduce colormap flashing on low-end 8-bit frame buffers by providing a policy for sharing; and c) when colormaps aren't shared, define a behavior in the X server color allocation scheme to reduce colormap flashing.

To encourage colormap sharing and accommodate special colormap requirements two new protocols are defined: the first provides a way to query the server for a list of reserved colormap entries, and the second is a way to initialize read-only (sharable) colormap entries at specific locations in a colormap.

To minimize colormap flashing when the root window's default visual is one of GrayScale, PseudoColor, or DirectColor, and a private colormap for the default visual is being used, a minor (but compatible) change to the server implementation of the AllocColor and AllocNamedColor requests is required. Where the core protocol says nothing about the pixel values returned, when this extension is in effect, the AllocColor and AllocNamedColor requests will first look for a matching color in the default colormap, and, if a match is found and the same cell in the private colormap has not already been allocated, the color will be allocated in the private colormap at the same location as in the default colormap (instead of in the first available location.)

14. Easy Resource Configuration

Setting and changing resources in X applications can be difficult for both the application programmer and the end user. **Resource Configuration Management (RCM)** addresses this problem by changing the **X Intrinsics** to immediately modify a resource for a specified widget and each child widget in the hierarchy. In this context, immediate means: no sourcing of a resource file is required; the application does not need to be restarted for the new resource values to take effect; and the change occurs immediately.

The main difference between **RCM** and the **Editres** protocol is that the **RCM** customizing hooks reside in the **Intrinsics** and thus are linked with other toolkits such as Motif and the Athena widgets. However, the **EditRes** protocol requires the application to link with the **EditRes** routines in the Xmu library and Xmu is not used by all applications that use Motif.

Easy Resource Configuration is not a standard part of the X Toolkit Intrinsics (libXt). It is neither an X Consortium standard nor an X Project Team specification.

15. Xinerama

The Xinerama extension provides a way for a multi-headed system to function as one large screen. Windows can span multiple screens and can move from one screen to another.

Currently, the Xinerama Extension works in a homogeneous graphics environment. A graphics environment is considered homogeneous if, for example, all of the graphics cards have 8 planes with 6 visuals. Mixing a 24-plane graphics card with a 8-plane card creates a heterogeneous environment.

Unlike other multiple screen implementations, Xinerama provides a solution at the device-independent level. The advantage of this approach is that it reduces the amount of work involved in supporting and maintaining the extension. The number of graphics devices on the market continues to grow; embedding the extension functionality into the device dependent code for each device would be a maintenance nightmare. Since the Xinerama implementation does not require any low-level graphics modifications, existing device-dependent code does not have to be recompiled. In the loadable server world, the Xinerama Extension will work with existing device-dependent shared libraries.

The Xinerama extension is not a standard. It is neither an X Consortium standard nor an X Project Team specification.

16. ANSIfication

R6.1 was officially the last release that supported traditional K&R C. Like R6.3, R6.4 assumes a Standard C compiler and environment. We have not intentionally removed any K&R C support from old code, and most of the release will continue to build on platforms without an ANSI C compiler.

17. VSW5

We have tested this release with VSW5 version 5.0.0. This release passes all tests in VSW5 with the following exceptions:

- tests for which a permanent waiver has been granted.
- tests for which a temporary waiver have been granted.
- tests where a defect in the test has been identified and reported.

VSW licensees may obtain a list of waivers granted from http://www.rdg.opengroup.org/interpretations/database/.

18. Year 2000 (Y2K) Compliance

For a statement of compliance see http://www.camb.opengroup.org/tech/desktop/faq/y2k.htm

19. Memory Testing

Beginning circa X11R5 the MIT X Consortium staff, and later the X Consortium, Inc. staff, and now the X Project Team staff have routinely tested this implementation for a variety of memory-type errors such as leaks, array bounds writes, uninitialized memory reads, and a variety of other errors; using a combination of commercial and "home grown" memory testing tools. All the real problems were fixed long ago; however we aren't so naive as to believe that there no remaining bugs. If you find a memory problem in this implementation please file a bug-report. If you find a memory problem in your vendor's implementation, tell your vendor. The popular commercial memory checking tools emit lots of false or spurious warnings, most of which can be safely ignored.

20. Security Considerations

On UNIX and UNIX-like operating systems there are serious security implications associated with running suid-root programs.

By default the xterm terminal emulation program is installed suid-root in order to be able to update utmp or utmpx entries. All the known (as of this writing) exploitable security holes in the X libraries have been eliminated — making it theoretically safe for xterm to be suid-root. For additional security you may install xterm without suid-root; however if you do, xterm will not be able to make utmp or utmpx entries.

On many Intel-based machines the X server must have root privileges in order to access the graphics card and open other devices. The easiest way to grant the requisite privileges is to use xdm to run your X server. Some people, who prefer not to use xdm, often work around the need for the X server to run with root privileges by making their X server a suid-root program. While all the known (as of this writing) exploitable security holes in the server have been eliminated, the X Project Team still recommends that you **not** make your X server suid-root. There are *safe* suid-root wrapper programs available (but not in this release) that you can use to start your server if you don't want to use xdm.

21. Filing Bug Reports

If you find a reproducible bug in software built from the source in this distribution or find bugs in its documentation, please complete a bug-report using the form in the file **xc/bug-report** and send it to The Open Group X Project Team at

mailto:xbugs@opengroup.org

Please try to provide all of the information requested on the form if it is applicable; the little extra time you spend on the report will make it much easier for someone to reproduce, find, and fix the bug.

Bugs in the contributed software that is available on the net are not handled on any official basis. Consult the documentation for the individual software to see where (if anywhere) to report the bug.

22. Acknowledgements

Release 6.4 of X11 was brought to you by the X Project Team staff at The Open Group: Arthur Barstow, Kaleb Keithley, Sekhar Makkapati, M.S. Ramesh, Jingping Ge, Ken Flowers, and Dave Knorr.

Several companies and individuals have cooperated and worked hard to make this release a reality, and our thanks go out to them:

Madeline Asmus of Digital for Xinerama.

Peter Daifuku of Silicon Graphics for Extended-Visual-Information.

Scott Revelt of Sun Microsystems for preliminary work on TOG-CUP.

Rob Lembree, formerly of Digital, for DPMS.

Jeff Walls of Hewlett Packard.

Wojtek Jarosz of Attachmate.

Bob Schulman of Seaweed.

Brian Bobryk of Digital.

Tom Brown of NetManage.

Garry Paxinos of Metro Link.

Victor Gold of Peritek.

Jackie Drane of IBM.

Table of Contents