

Release Notes for X11R6.8.2

The X.Org Foundation
The XFree86 Project, Inc.

9 February 2005

Abstract

These release notes contains information about features and their status in the X.Org Foundation X11R6.8.2 release. It is based on the XFree86 4.4RC2 RELNOTES document published by The XFree86™ Project, Inc. There are significant updates and differences in the X.Org release as noted below.

1. Introduction to the X11R6.8.2 Release

The release numbering is based on the original MIT X numbering system. X11 refers to the version of the network protocol that the X Window system is based on: Version 11 was first released in 1988 and has been stable for 15 years, with only upward compatible additions to the core X protocol, a record of stability envied in computing. Formal releases of X started with X version 9 from MIT; the first commercial X products were based on X version 10. The MIT X Consortium and its successors, the X Consortium, the Open Group X Project Team, and the X.Org Group released versions X11R3 through X11R6.6, before the founding of the X.Org Foundation.

There will be future maintenance releases in the X11R6.8.x series. However, efforts are well underway to split the X distribution into its modular components to allow for easier maintenance and independent updates. We expect a transitional period while both X11R6.8 releases are being fielded and the modular release completed and deployed while both will be available as different consumers of X technology have different constraints on deployment. We have not yet decided how the modular X releases will be numbered.

We encourage you to submit bug fixes and enhancements to bugzilla.freedesktop.org using the xorg product, and discussions on this server take place on [<xorg@freedesktop.org>](mailto:xorg@freedesktop.org).

X11R6.8.2 comes with a graphical configuration tool called "xorgcfg", which also has a text mode interface and can be used to create an initial configuration file. It can also be used to customise existing configurations.

Next in the order of configuration preferences is to use the Xorg server's ability to create a starting configuration file. Run as root:

```
Xorg -configure
```

and follow the instructions.

Finally, if all else fails, the trusty old standby text-based tool "xorgconfig" can also be used for generating X server config files.

At least one, and hopefully, all of these configuration options will give you a reasonable starting point for a suitable configuration file. With the automatic mechanism you might even find that you don't need one!

If you do need to customise the configuration file, see the `xorg.conf` manual page. You can also check the driver-specific manual pages and the related documentation (found at *tables below* (section , page) also).

Before downloading the binary distributions for this release, please have a quick read through the Installation Document. It may save you some time and also help you figure out which of the binary releases you need.

The next section describes what is new in the latest version (6.8.2) compared with the previous full release (6.8.1). The other sections below describe some of the new features and changes between 3.3.x and 4.0. There are lots of new features, and we definitely don't have enough space to cover them all here.

2. Summary of new features in X11R6.8.2

This is a sampling of the new features in X11R6.8.2. A more complete list of changes can be found in the ChangeLog file that is part of the X source tree.

- Freetype was updated to version 2.1.8. But installing freetype from X distributions would often or usually result in the replacement or use of "stale" versions of freetype. On Linux, FreeBSD, Solaris 10, and SCO5, therefore, X11R6.8.2 will by default use the version of freetype2 that is installed on the system. If your system doesn't come with an installed freetype2 and you wish to use the version supplied with this distribution, please add:

```
#define HasFreetype2 NO
```

```
to config/cf/host.def.
```

- The XTT font module and FreeType1 have been retired because FreeType2 subsumes their functionality
- Retire old PS Type1 font rasterizer (except for CID font usage) since the FreeType2 rasterizer now handles PS Type1 (*.pfa, *.pfb)
- Render implementation fixes
- Updated x86emu and resynced with upstream at Scitech
- Updated SiS driver
- Updated Nvidia driver (opensource version)
- Render acceleration for ATI's R100 and R200-series cards
- Substantial speedups in the software implementation of the render extensions when compiled with gcc 3.4 on the i386 architecture.
- Infrastructure for rotation support in drivers
- New Trapezoid specification for the Render extension
 - Respecify Render to include only 'normal' traps
 - Allow backward compatibility but internally covert to new format
- Software mouse cursor is now based on the Damage extension
- A new keyboard driver is enabled by default. The old driver is disabled unless explicitly compiled in by defining the macro `|USE_DEPRECATED_KEYBOARD_DRIVER|`.

- All extensions (except Xserver-specific extensions "DMX" and "XpExtension") can now be enabled/disabled from the configuration file and from the command line.
- Mac OS X updates:
 - Support dynamic screen configuration changes in rootless mode
 - Added option to always use Mac command key equivalents
 - Interpret scroll wheel mouse events correctly when shift is held down
 - Added trivial Xinput support
 - Fixed launch of X clients from Finder with a space in their path
 - Fixed some GLX rendering problems on Mac OS X 10.2 and earlier
- Updated xterm version

2.1 New X extensions

X11R6.8.2 includes four new extensions:

XFixes

The XFixes extension is a collection of improvements for deficiencies in the core protocol, including

- Notification when a selection changes
- Server-side region objects
- Allow clients to track the cursor image

Damage

The Damage extension allows a client to be notified whenever something is drawn to a window. This feature is useful for VNC servers, for screen magnifiers, and for clients using the Composite extension to update the screen.

Composite (experimental)

The Composite extension allows a client to request that all drawing to window is redirected to off-screen buffer. Though the Damage extension the client, called a 'compositing manager', can know which areas of a window is modified and render the windows on screen. By making use of the drawing requests from both the core protocol and the RENDER extension, the compositing manager can create special effects, such as translucent windows.

The Composite extension is considered experimental in X11R6.8.2 and is turned off by default.

XEvIE (X Event Interception Extension) (experimental)

XEvIE is an extension to intercept core keyboard and pointing device input. It allows consumption, modification or synthesis of input events before these are sent to their final destination (i.e., interested clients). This feature is required by the GNOME accessibility project.

The XEvIE extension is considered experimental in X11R6.8.2 and is turned off by default.

As noted above, some extensions are disabled by default. They can be enabled either in the `xorg.conf` configuration file or on the `Xorg` command-line. For example, to enable the Composite extension, you could include a section like the following in the `xorg.conf` file:

```
Section "Extensions"  
    Option "Composite" "Enable"  
EndSection
```

or you could put `+extension Composite` on your command line.

2.2 Distributed Multihead X (DMX)

X11R6.8.2 includes a new X server, called `Xdmx`, that allows users to create a single unified desktop from multiple running X servers. It works by creating a local screen on each of these running X servers, which can be distributed across a network, and then presents that set of screens to the user. When combined with `Xinerama`, a unified workspace, ranging from large display walls to small two display environments, can be created.

More information about DMX can be found on the DMX Project's web site [<URL:http://dmx.sf.net/>](http://dmx.sf.net/), including how to configure and use `Xdmx`.

2.3 Xprint updates

- Integrated various fixes from xprint.mozdev.org [<URL:http://xprint.mozdev.org>](http://xprint.mozdev.org) (more or less the whole codebase has been merged)
- Integrated various fixes from Sun Microsystems
- OpenGL is now supported for printing
- New set of maintenance tools for the Xprint server (`xplsprinters`, `xprehashprinterlist`, etc.)
- Added new `XprintUtils` client library to make the usage of Xprint easier
- Various other features and fixes integrated
- Athena toolkit now has print support (`XawPrintShell`) which is included in a new version of that library (`Xaw8`)
- Various applications/tools like `xman`, `xedit`, `xlogo`, `xlsfonts`, etc. now have print support

2.4 Updated Mesa and DRI from upstream sources

- Mesa version has been updated to 6.2.1 (in X11R6.8.2)
- More OpenGL extensions
- MergedFB for Radeon (Dualhead DRI support on these cards)
- Many GLX fixes
- Working SiS DRI driver
- Major Radeon and R200 DRI driver updates
- `fbconfigs` support
- Beginnings of `pbuffer` support (indirect only, and only in specific circumstances).
- Merge Mach64 DRI support (Eric Anholt, July 23)

2.5 Video driver enhancements

- ATI Radeon driver updates:
 - Merged Framebuffer support (dualhead with DRI)
 - `DynamicClocks` option (reduced power usage)

- Render acceleration (r100, r200 chips only)
- Support for new ATI chips (R420/M18, R423, RV370/M22, RV380/M24, RS300)
- DRI support for IGP chips
- Xv gamma correction
- Updated 3D drivers
- Many other small fixes
- Chips driver update
 - Improved BE support
- MGA driver updates
 - Support for DDC and DPMS on second head on G400
 - Updated 3D driver
- Neomagic driver updates
 - Support for Xv on pre-nm2160 chips
 - Pseudocolor overlay mode (=PseudoColor emulation)
 - Improved support for lowres double scan modes
- i810 driver updates
 - Dualhead support (i830+)
 - i915 support
 - New 3D driver (i830+)
 - i810 driver is now supported for AMD64
- S3 driver updates
 - Support for additional IBM RAMDACs
- Savage driver updates
 - Pseudocolor overlay mode
- SiS driver updates include
 - output device hotplugging
 - lots of fixes for 661, 741, 760
 - extended interface for SiSCtrl?
 - extended LCD handling (allow more modes)
 - HDTV support (480p, 480i, 720p, 1080i; 315/330 series)
 - Added video blitter Xv adapter (315/330 series)
 - extended RENDER acceleration
 - SiS driver now supported on AMD64
- New Voodoo driver (Alan Cox)
 - Provides native (glide-less) acceleration and mode setup for voodoo/voodoo2 boards

3. Drivers

3.1 Video Drivers

X11R6.8.2 includes the following video drivers:

Driver Name	Description	Further Information
apm	Alliance Pro Motion	README.apm
ark	Ark Logic	
ati	ATI	README.ati, README.r128, r128(4), radeon(4)
chips	Chips & Technologies	README.chips, chips(4)
cirrus	Cirrus Logic	
cyrix (*)	Cyrix MediaGX	README.cyrix
fbdev	Linux framebuffer device	fbdev(4)
glide	Glide2x (3Dfx)	glide(4)
glint	3Dlabs, TI	glint(4)
i128	Number Nine	README.I128, i128(4)
i740	Intel i740	README.i740
i810	Intel i8xx	README.i810, i810(4)
imstt	Integrated Micro Solns	
mga	Matrox	mga(4)
neomagic	NeoMagic	neomagic(4)
newport (-)	SGI Newport	README.newport, newport(4)
nsc	National Semiconductor	nsc(4)
nv	NVIDIA	nv(4)
rendition	Rendition	README.rendition, rendition(4)
s3	S3 (not ViRGE or Savage)	
s3virge	S3 ViRGE	README.s3virge, s3virge(4)
savage	S3 Savage	savage(4)
siliconmotion	Silicon Motion	siliconmotion(4)
sis	SiS	README.SiS, sis(4)
sunbw2 (+)	Sun bw2	
suncg14 (+)	Sun cg14	
suncg3 (+)	Sun cg3	
suncg6 (+)	Sun GX and Turbo GX	
sunffb (+)	Sun Creator/3D, Elite 3D	
sunleo (+)	Sun Leo (ZX)	
suntcx (+)	Sun TCX	
tdfx	3Dfx	tdfx(4)
tga	DEC TGA	README.DECtga
trident	Trident	trident(4)
tseng	Tseng Labs	
via	VIA	via(4)
vesa	VESA	vesa(4)
vga	Generic VGA	vga(4)
vmware	VMWare guest OS	vmware(4)

Drivers marked with (*) are present in a preliminary form in this release, but are not complete and/or stable yet.

Drivers marked with (+) are for Linux/Sparc only.

Drivers marked with (-) are for Linux/mips only.

Darwin/Mac OS X uses IOKit drivers and does not use the module loader drivers listed above. Further information can be found in README.Darwin.

X11R6.8.2 includes the following print drivers:

3.2 Print Drivers

Driver Name	Description	Further Information
PostScript	PostScript driver	
PCL	PCL color driver	
PCL-MONO	PCL monochrome driver	
RASTER	Monochrome raster driver (XWD, PS, PCL)	

Drivers marked with (*) are available for Linux only.

X11R6.8.2 includes the following input drivers:

3.3 Input Drivers

Driver Name	Description	Further Information
aiptek (*)	Aiptek USB tablet	aiptek(4)
calcomp	Calcomp	
citron	Citron	citron(4)
digitaledge	DigitalEdge	
dmc	DMC	dmc(4)
dynapro	Dynapro	
elographics	EloGraphics	
fpit	Fujitsu Stylistic Tablet PCs	fpit(4)
hyperpen	Aiptek HyperPen 6000	
js_x	JamStudio pentablet	js_x(4)
kbd	generic keyboards (loadable module)	kbd(4)
keyboard	generic keyboards (builtin driver)	keyboard(4)
microtouch	MicroTouch	
mouse	most mouse devices	mouse(4)
mutouch	MicroTouch	
palmax	Palmax PD1000/PD1100	palmax(4)
penmount	PenMount	
spaceorb	SpaceOrb	
summa	SummaGraphics	
tek4957	Tektronix 4957 tablet	tek4957(4)
ur98 (*)	Union Reality UR-F98 headtracker	ur98(4)
void	dummy device	void(4)
wacom	Wacom tablets	wacom(4)

Drivers marked with (*) are available for Linux only.

4. Overview of X11R6.8.2

On most platforms, X11R6.8.2 has a single X server binary called `Xorg`. This binary can either have one or more video and input drivers linked in statically, or more usually, dynamically, and in that manner load the video drivers, input drivers, and other modules that are needed.

X11R6.8.2 has X server support for most UNIX® and UNIX-like operating systems on Intel/x86 platforms, plus support for Linux and some BSD OSs on Alpha, PowerPC, IA-64, AMD64, Sparc, and Mips platforms, and for Darwin on PowerPC.

4.1 Loader and Modules

The X server has a built-in run-time loader, which can load normal object files and libraries in most of the commonly used formats. The loader does not rely on an operating system's native dynamic loader support and it works on platforms that do not provide this feature. This allows for the modules to be operating system independent (although not, of course, CPU architecture independent) which means that a module compiled on Linux/x86 can be loaded by an X server running on Solaris/x86, or FreeBSD, or even OS/2.

A main benefit of this, is that when modules are updated, they do not need to be recompiled for every different operating system. The loader in version 6.8.2 has support for Intel (x86), Alpha and PowerPC platforms. It also has preliminary support for Sparc platforms.

The X server makes use of modules for video drivers, X server extensions, font rasterisers, input device drivers, framebuffer layers (like mfb, cfb, etc), and internal components used by some drivers (like XAA),

The module interfaces (both API and ABI) used in this release are subject to change without notice. While we will attempt to provide backward compatibility for the module interfaces as of the 4.0 release (meaning that 4.0 modules will work with future core X server binaries), we cannot guarantee this. Compatibility in the other direction is explicitly not guaranteed because new modules may rely on interfaces added in new releases.

Note about module security

The X server runs with root privileges, i.e., the X server loadable modules also run with these privileges. For this reason we recommend that all users be careful to only use loadable modules from reliable sources, otherwise the introduction of viruses and contaminated code can occur and wreak havoc on your system. We hope to have a mechanism for signing/verifying the modules that we provide available in a future release.

4.2 Configuration File

The X server uses a configuration file as the primary mechanism for providing configuration and run-time parameters. The configuration file format is described in detail in the `xorg.conf(5)` manual page.

The X server has support for automatically determining an initial configuration on most platforms, as well as support or generating a basic initial configuration file.

4.3 Command Line Options

Command line options can be used to override some default parameters and parameters provided in the configuration file. These command line options are described in the `Xorg(1)` manual page.

4.4 XAA

The XFree86 Acceleration Architecture (XAA) was completely rewritten from scratch for XFree86 4.x and is used in X11R6.8.2. Most drivers implement acceleration by making use of the XAA module. The Xorg server will accept modules built either for XFree86 4.4 servers or its own.

4.5 Multi-head

Some multi-head configurations are supported in X11R6.8.2, primarily with multiple PCI/AGP cards.

One of the main problems is with drivers not sufficiently initialising cards that were not initialised at boot time. This has been improved somewhat with the INT10 support that is used by most drivers (which allows secondary card to be "soft-booted", but in some cases there are other

issues that still need to be resolved. Some combinations can be made to work better by changing which card is the primary card (either by using a different PCI slot, or by changing the system BIOS's preference for the primary card).

4.6 Xinerama

Xinerama is an X server extension that allows multiple physical screens to behave as a single screen. With traditional multi-head in X11, windows cannot span or cross physical screens. Xinerama removes this limitation. Xinerama does, however, require that the physical screens all have the same root depth, so it isn't possible, for example, to use an 8-bit screen together with a 16-bit screen in Xinerama mode.

Xinerama is not enabled by default, and can be enabled with the `+xinerama` command line option for the X server.

Xinerama was included with X11R6.4. The version included in X11R6.8.2 was completely rewritten for improved performance and correctness.

Known problems:

- Most window managers are not Xinerama-aware, and so some operations like window placement and resizing might not behave in an ideal way. This is an issue that needs to be dealt with in the individual window managers, and isn't specifically an X server problem.

4.7 DGA version 2

DGA 2.0 is included in 6.8.2. Documentation for the client libraries can be found in the XDGA(3) man page. A good degree of backwards compatibility with version 1.0 is provided.

4.8 DDC

The VESA® Display Data Channel (DDC™) standard allows the monitor to tell the video card (or on some cases the computer directly) about itself; particularly the supported screen resolutions and refresh rates.

Partial or complete DDC support is available in most of the video drivers. DDC is enabled by default, but can be disabled with a "Device" section entry: `Option "NoDDC"`. We have support for DDC versions 1 and 2; these can be disabled independently with `Option "NoDDC1"` and `Option "NoDDC2"`.

At startup the server prints out DDC information from the display, and can use this information to set the default monitor parameters, or to warn about monitor sync limits if those provided in the configuration file don't match those that are detected.

4.8.1 Changed behavior caused by DDC.

Several drivers uses DDC information to set the screen size and pitch. This can be overridden by explicitly resetting it to the and non-DDC default value 75 with the `-dpi 75` command line option for the X server, or by specifying appropriate screen dimensions with the "DisplaySize" keyword in the "Monitor" section of the config file.

4.9 GLX and the Direct Rendering Infrastructure (DRI)

Direct rendered OpenGL® support is provided for several hardware platforms by the Direct Rendering Infrastructure (DRI). Further information about DRI can be found at the DRI Project's web site [<URL:http://dri.sf.net/>](http://dri.sf.net/). The 3D core rendering component is provided by Mesa [<URL:http://www.mesa3d.org>](http://www.mesa3d.org).

4.10 XVideo Extension (Xv)

The XVideo extension is supported in X11R6.7.x and newer releases. An `XvQueryPortAttributes` function has been added as well as support for `XvImages`. `XvImages` are `XImages` in alternate color spaces such as YUV and can be passed to the server through shared memory segments. This allows clients to display YUV data with high quality hardware scaling and filtering.

4.11 X Rendering Extension (Render)

The X Rendering extension provides a 2D rendering model that more closely matches application demands and hardware capabilities. It provides a rendering model derived from Plan 9 based on Porter/Duff image composition rather than binary raster operations.

Using simple compositing operators provided by most hardware, Render can draw anti-aliased text and geometric objects as well as perform translucent image overlays and other image operations not possible with the core X rendering system.

Unlike the core protocol, Render provides no font support for applications, rather it allows applications to upload glyphs for display on the screen. This allows the client greater control over text rendering and complete access to the available font information while still providing hardware acceleration. The Xft library provides font access for Render applications.

4.11.1 The Xft Library

On the client side, the Xft library provides access to fonts for applications using the FreeType library, version 2. One important thing to note is that Xft uses the vertical size of the monitor to compute accurate pixel sizes for provided point sizes; if your monitor doesn't provide accurate information via DDC, you may want to add that information to `xorg.conf`.

To allow a graceful transition for applications moving from core text rendering to the Render extension, Xft can use either the core rendering requests or the Render extension for text. See the section on FreeType support in Xft for instructions on configuring X11R6.8.2 to use an existing FreeType installation.

The Xft library uses configuration files, `/etc/fonts/fonts.conf` and `/etc/fonts/local.conf`, which contains information about which directories contain font files and also provides a sophisticated font aliasing mechanism. Documentation for that file is included in the `Xft(3)`, `fontconfig(3)` and `fonts.conf(4)` man pages.

4.11.2 Application Support For Anti-Aliased Text

Only four applications have been modified in X11R6.8.2 to work with the Render extension and the Xft and FreeType libraries to provide anti-aliased text: `xterm`, `xditview`, `x11perf` and `xclock`. Migration of other applications may occur in future releases.

By default, `xterm` uses core fonts through the standard core API. It has a command line option and associated resource to direct it to use Xft instead:

- `-fa family / .VT100.faceName: family`. Selects the font family to use.

`Xditview` will use Xft instead of the core API by default. `X11perf` includes tests to measure the performance of text rendered in three ways, anti-aliased, anti-aliased with sub-pixel sampling and regular chunky text, but through the Render extension, a path which is currently somewhat slower than core text.

`Xclock` uses the Render extension to draw the analog face and shares the `-fa` option and `faceName` resources with `xterm` to select a font for the digital mode.

4.12 Other extensions

The XFree86-Misc extension has not been fully ported to the new server architecture yet. This should be completed in a future release.

The XFree86-VidModeExtension extension has been updated, and mostly ported to the new server architecture. The area of mode validation needs further work, and the extension should be used with care. This extension has support for changing the gamma setting at run-time, for modes where this is possible. The `xgamma` utility makes use of this feature. Compatibility with the 3.3.x version of the extension is provided. The missing parts of this extension and some new features should be completed in a future release.

4.13 xedit

Xedit has several new features, including:

- An embedded lisp interpreter that allows easier extension of the editor.
- Several new syntax highlight modes, and indentation rules for C and Lisp.
- Flexible search/replace interface that allows regex matches.
- Please refer to `xedit(1)` for more details.
- Support for printing via Xprint.

4.14 Font support

Details about the font support in X11R6.8.x can be found in the `README.fonts` document.

4.15 TrueType support

X11R6.7 came with two TrueType backends. The functionality from the 'X-TrueType' ("XTT") backend has been integrated into the 'FreeType' backend which is designed to transparently support all of the functionality from the 'X-TrueType' backend with the exception of the font encoding libraries; the 'FreeType' backend uses only the fontenc-based encoding system.

4.16 CID font support

Support for CID-keyed fonts is included in X11R6.8.2 The CID-keyed font format was designed by Adobe Systems <[URL:http://www.adobe.com](http://www.adobe.com)> for fonts with large character sets. The CID-keyed font support in X11R6.8.2 was donated by SGI <[URL:http://www.sgi.com](http://www.sgi.com)>. See the LICENSE document for a copy of the CID Font Code Public License.

4.17 Internationalisation of the scalable font backends

X11R6.8.2 has a "fontenc" layer to allow the scalable font backends to use a common method of font re-encoding. This re-encoding makes it possible to use fonts in encodings other than their native encoding. This layer is used by the Type1, Speedo and FreeType backends.

4.18 Large font optimisation

The glyph metrics array, which all the X clients using a particular font have access to, is placed in shared memory, so as to reduce redundant memory consumption. For non-local clients, the glyph metrics array is transmitted in a compressed format.

4.19 Unicode/ISO 10646 support

What is included in X11R6.8.2

- All "-misc-fixed-*" BDF fonts are now available in the ISO10646-1 encoding and cover at least the 614 characters found in ISO 8859-{1-5,7-10,14,15}, CP1252, and MES-1. The non-bold fonts also cover all Windows Glyph List 4 (WGL4) characters, including those found in

all 8-bit MS-DOS/Windows code pages. The 8-bit variants of the “-misc-fixed-” BDF fonts (ISO8859-1, ISO8859-2, KOI8-R, etc.) have all been automatically generated from the new ISO10646-1 master fonts.

- Some “-misc-fixed-” BDF ISO10646-1 fonts now cover a comprehensive Unicode repertoire of over 3000 characters including all Latin, Greek, Cyrillic, Armenian, Gregorian, Hebrew, IPA, and APL characters, plus numerous scientific, typographic, technical, and backwards-compatibility symbols. Some of these fonts also cover Arabic, Ethiopian, Thai, Han/Kanji, Hangul, full ISO 8859, and more. For the 6x13 font there is now a 12x13ja Kanji extension and for the 9x18 font there is a 18x18ja Kanji/Han/Hangul extension, which covers all ISO-2022-JP-2 (RFC 1554) characters. The 9x18 font can also be used to implement simple combining characters by accent overstriking. For more information, read Markus Kuhn’s UTF-8 and Unicode FAQ <URL: <http://www.cl.cam.ac.uk/~mgk25/unicode.html>>.
- Mark Leisher’s ClearlyU proportional font (similar to Computer Modern).
- ISO 10646/Unicode UTF-8 Level 1 support added to xterm (enabled with the `-u8` option).
- The Freetype backend (the “`freetype`” module) supports Unicode-encoded fonts.

4.20 Xlib Compose file support and extensions

A more flexible Compose file processing system was added to Xlib in X11R6.8.2. The compose file is searched for in the following order:

1. If the environment variable `$XCOMPOSEFILE` is set, its value is used as the name of the Compose file.
2. If the user’s home directory has a file named “.XCompose”, it is used as the Compose file.
3. The old method is used, and the compose file is “<localdir>/<localname>/Compose”.

Compose files can now use an “include” instruction. This allows local modifications to be made to existing compose files without including all of the content directly. For example, the system’s iso8859-1 compose file can be included with a line like this:

```
include "/usr/X11R6/lib/X11/locale/iso8859-1/Compose"
```

There are two substitutions that can be made in the file name of the include instruction. `%H` expands to the user’s home directory (the `$HOME` environment variable), and `%L` expands to the name of the locale specific Compose file (i.e., “<localdir>/<localname>/Compose”).

For example, you can include in your compose file the default Compose file by using:

```
include "%L"
```

and then rewrite only the few rules that you need to change. New compose rules can be added, and previous ones replaced.

Finally, it is no longer necessary to specify in the right part of a rule a locale encoded string in addition to the keysym name. If the string is omitted, Xlib figures it out from the keysym according to the current locale. I.e., if a rule looks like:

```
<dead_grave> <A> : "\300" Agrave
```

the result of the composition is always the letter with the “\300” code. But if the rule is:

```
<dead_grave> <A> : Agrave
```

the result depends on how Agrave is mapped in the current locale.

4.21 Bitstream Vera fonts

X11R6.8.2 includes the Bitstream Vera family of typefaces in TrueType format. This family includes the "Bitstream Vera Sans", "Bitstream Vera Sans Mono" and "Bitstream Vera Serif" in Roman and Bold variants as well as the "Bitstream Vera Sans" and "Bitstream Vera Sans Mono" in Oblique and Bold Oblique. These fonts include all of the glyphs needed for ISO 8859 parts 1 9 and 15.

The license terms for the Vera fonts are included in the file `COPYRIGHT.Vera`.

4.22 Luxi fonts from Bigelow and Holmes

The X distribution includes the "Luxi" family of Type 1 fonts and TrueType fonts. This family consists of the fonts "Luxi Serif", "Luxi Sans" and "Luxi Mono" in Roman, oblique, bold and bold oblique variants. The TrueType version have glyphs covering the basic ASCII Unicode range, the Latin 1 range, as well as the *Extended Latin* range and some additional punctuation characters. In particular, these fonts include all the glyphs needed for ISO 8859 parts 1, 2, 3, 4, 9, 13 and 15, as well as all the glyphs in the Adobe Standard encoding and the Windows 3.1 character set.

The glyph coverage of the Type 1 versions is somewhat reduced, and only covers ISO 8859 parts 1, 2 and 15 as well as the Adobe Standard encoding.

The Luxi fonts are original designs by Kris Holmes and Charles Bigelow from Bigelow and Holmes Inc., who developed the Luxi typeface designs in Ikarus digital format. URW++ Design and Development GmbH converted the Ikarus format fonts to TrueType and Type 1 font programs and implemented the grid-fitting "hints" and kerning tables in the Luxi fonts.

The license terms for the Luxi fonts are included in the file 'COPYRIGHT.BH', as well as in the License document. For further information, please contact <design@bigelowandholmes.com> or <info@urwpp.de>, or consult the URW++ web site <URL:http://www.urwpp.de>.

5. Miscellaneous

This section describes other items of note for the X11R6.8.2 release.

5.1 Legacy keyboard driver phase-out

The legacy keyboard driver is no longer compiled into the X server by default on certain platforms (including Linux). The newer `kbd` driver replaces the older built-in driver. It is suggested that, if the X server says that it cannot load the keyboard driver, then the `xorg.conf` file should be updated to use the new `kbd` driver, which can be done by changing the `Driver` line in the `InputDevice` section. For example,

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver      "kbd"
EndSection
```

Note that the driver name is case-sensitive.

5.2 Socket directory ownership and permissions

The socket directories created in `/tmp` are now required to be owned by root and have their sticky-bit set. If the permissions are not set correctly, the component using this directory will print an error message and fail to start. Common socket directories that are known to be affected include:

```
/tmp/.font-unix
/tmp/.ICE-unix
/tmp/.X11-unix
```

These directories are used by the font server, `xfs`, applications using the Inter-Client Exchange protocol (ICE) and the X server, respectively.

There are several solutions to the problem of when to create these directories. They could be created at install time by the system's installer if the `/tmp` dir is persistent. They could be created at boot time by the system's boot scripts (e.g., the `init.d` scripts). Or, they could be created by PAM modules at service startup or user login time.

The solution chosen is platform dependent, and the system administrator should be able to handle creating those directories on any systems that do not have the correct ownership or permissions.

5.3 Composite exposes extra visuals

When the Composite extension is enabled via `xorg.conf` or the command line, a new visual is created. This visual is different from the other visuals used by X applications in that it includes an alpha component. It is used by the compositing manager and other Composite aware applications.

Most X applications ignore this visual since it is not useful to them; however some applications mistakenly try to use it, which will cause them to fail. An environment variable, `XLIB_SKIP_ARGB_VISUALS`, was added to the X11 library to hide this visual from applications that mistakenly try to use it. If an application fails only when the Composite is enabled, try setting this environment variable before starting the application.

Since Composite is not enabled by default, it is not expected that this issue will be visible to most users.

6. Attributions/Acknowledgements/Credits

This section lists the credits for the X11R6.8.2 release. For a more detailed breakdown, refer to the `ChangeLog` file in the X.Org source tree, the `ChangeLog's` in or the 'cvs log' information for individual source files."

These people contributed in some way to X11R6.8.2

Paul Anderson, Eric Anholt, Dave Airlie, Donnie Berkholz, Daniel Berrange, Thomas Biege, Russ Blaine, Ryan Breen, Alan Coopersmith, Jay Cotton, Michel Daenzer, David Dawes, Kevin DeKorte, Alex Deucher, Stefan Dirsch, Egbert Eich, Rik Faith, Jim Gettys, Giuseppe Ghibò, Thomas Gilg, Alexander Gottwald, Mike A. Harris, John Harper, John Heasley, Jeremy Huddleston, Matthieu Herrb, Alan Hourihane, Jay Hobson, Matthias Hopf, Kristian Høvsberg, Harold L. Hunt II, Adam Jackson, Daniel Jacobowitz, Deron Johnson, Masaki Katakai, Stephen Kennedy, Ivan Kokshaysky, Stuart Kreitman, Peter Kunzman, Julien Lafon, Nolan Leake, Ryan Lortie, Andreas Luik, Torrey T. Lyons, Roland Mainz, Guy Martin, Kevin E. Martin, Dan McNichol, Dagfinn Ilmari Mannsaker, Simon Montagu, Keith Packard, Greg Parker, Drew Parsons, Aaron Plattner, Branden Robinson, Søren Sandmann, Ty Sarna, Yu Shao, Gustavo Noronha Silva, Bryan Stine, Owen Taylor, Travis Tilley, Ryan Underwood, Ronny Vindenes, Chisato Yamauchi, Mark Vojkovich, Tim Yamin, Hui Yu

The X Window System has been a collaborative effort from its inception. Our apologies for anyone or organization inadvertently overlooked. Many individuals (including major contributors) who worked on X are represented by their employers in this list.

This product includes software developed by:

Paul Anderson, Michael Bax, Jehan Bing, Peter Breitenlohner, Alan Coopersmith, Thomas Dickey, Egbert Eich, John Dennis, Fabrizio Gennari, Jim Gettys, Alexander Gottwald, Ralf Habacker, Mike Harris, Mattheiu Herrb, Alan Hourihane, Harold L Hunt II, Elliot Lee, Jeremy Katz, Kaleb Keithley, Stuart Kreitman, Andreas Luik, Torrey Lyons, Roland Mainz, Kevin E. Martin, Takuma Murakami, Kensuke Matsuzaki, Keith Packard, Ivan Pascal, Earle F. Philhower III, Benjamin Riefenstahl, Leon Shiman, Toshimitsu Tanaka, Nicholas Wourms, 2d3d Inc., Aaron Plattner, Adam de Boor, Adam Jackson, Adobe Systems Inc., After X-TT Project, AGE Logic Inc., Alan Coopersmith, Alan Cox, Alan Hourihane, Alexander Gottwald, Alex Deucher, Andreas Luik, Andreas Monitzer, Andrew C Aitchison, Andy Ritger, Ani Joshi, Anton Zioviev, Apollo Computer Inc., Apple Computer Inc., Ares Software Corp., AT&T Inc., ATI Technologies Inc., Benjamin Riefenstahl, Bigelow and Holmes, Bill Reynolds, Bitstream Inc., Bitstream, Inc, Brian Fundakowski Feldman, Brian Goines, Brian Paul, Bruno Haible, Bryan Stine, Charles Murcko, Chen Xiangyang, Chisato Yamauchi, Chris Constello, Christian Zietz, Cognition Corp., Compaq Computer Corporation, Concurrent Computer Corporation, Conectiva S.A., Corin Anderson, Craig Struble, Daewoo Electronics Co. Ltd., Dale Schumacher, Damien Miller, Daniel Berrange, Daniel Stone, Daniver Limited, Daryll Strauss, Data General Corporation, David Bateman, David Dawes, David E. Wexelblat, David Holland, David J. McKay, David McCullough, David Mosberger-Tang, David S. Miller, Davor Matic, Deron Johnson, Digital Equipment Corporation, Dirk Hohndel, Doug Anson, Earle F. Philhower III, Edouard TISSERANT, Eduardo Horvath, Egbert Eich, Elliot Lee, Eric Anholt, Eric Fortune, Eric Sunshine, Erik Fortune, Erik Nygren, Evans & Sutherland Computer Corporation, Fabio Massimo Di Nitto Fabrizio Gennari, Finn Thøgersen, Frederic Lepied, Free Software Foundation Inc., Fujitsu Limited, Fujitsu Open Systems Solutions Inc., Fuji Xerox Co. Ltd., Geert Uytterhoeven, Gerrit Jan Akkerman, Gerry Toll, Glenn G. Lai, GNOME Foundation, Go Watanabe, Gregory Mokhin, Greg Parker, GROUPE BULL, Guy Martin, Hans Oey, Harald Koenig, Harm Hanemaayer, Harold L Hunt II, Harry Langenbacher, Henry A. Worth, Hewlett-Packard Company, Hitachi Ltd, Holger Veit, Howard Greenwell, Hummingbird Communications Ltd., IBM Corporation, Intel Corporation, INTERACTIVE Systems Corporation, International Business Machines Corp., Itai Nahshon, Ivan Kokshaysky, Ivan Pascal, Jakub Jelinek, James Tsillas, Jason Bacon, Jean-loup Gailly, Jeff Kirk, Jeffrey Hsu, Jehan Bing, Jeremy Katz, Jim Gettys, Jim Tsillas, J. Kean Johnston, John Dennis, John Harper, John Heasley, Jon Block, Jon Tombs, Jorge Delgado, Joseph Friedman, Joseph V. Moss, Juliusz Chroboczek, Jyunji Takagi, Kaleb Keithley, Kaleb S. Keithley, Kazushi (Jam) Marukawa, Kazuyuki (ikko-) Okamoto, Kean Johnston. Keith Packard, Keith Packard, Keith Whitwell, Kensuke Matsuzaki, Kristian Høgsberg, Larry Wall, Lawrence Berkeley Laboratory, Lennart Augustsson, Leon Shiman, Lexmark International Inc., Linus Torvalds, Luc Verhaegen, Machine Vision Holdings Inc., Manfred Brands, Marc Aurele La France Mark Adler, Mark J. Kilgard, Mark Leisher, Mark Smulders, Mark Vojkovich, Massachusetts Institute Of Technology, Matrox Graphics, Mattheiu Herrb Matthew Grossman, Matthieu Herrb, Metro Link Inc., Michael Bax, Michael H. Schimek, Michael P. Marking, Michael Schimek, Michael Smith, Michel Daenzer, Mike A. Harris, Mike Harris, Ming Yu, MIPS Computer Systems Inc., National Semiconductor, NCR Corporation Inc., Netscape Communications Corporation, Network Computing Devices Inc., Nicholas Wourms, Noah Levitt, Nolan Leake, Novell Inc., Nozomi YTOW, NTT Software Corporation, Number Nine Computer Corp., Number Nine Visual Technologies, NVIDIA Corp., Olivier Danet, Oki Technosystems Laboratory Inc., OMRON Corporation, Open Software Foundation, Orest Zborowski, Owen Taylor, Pablo Saratxaga, Panacea Inc., Panagiotis Tsirigotis, Paolo Severini, Pascal Haible, Patrick Lecoanet, Patrick Lerda, Paul Anderson, Paul Elliott, Peter Breitenlohner,

Peter Kunzmann, Peter Trattler, Philip Homburg, Precision Insight Inc., Prentice Hall, Quarterdeck Office Systems, Ralf Habacker Randy Hendry, Ranier Keller, Red Hat Inc., Regents of the University of California, Regis Cridlig, Rene Cougnenc, Richard A. Hecker, Richard Burdick, Rich Murphey, Rickard E. Faith, Rik Faith, Robert Baron, Robert Chesler, Robert Millan. Robert V. Baron, Robin Cutshaw, Roland Mainz, Ronny Vindenes, Russ Blaine, Ryan Breen, Ryan Lortie, Ryan Underwood, S3 Graphics Inc., Sam Leffler, SciTech Software, Scott Laird, Sebastien Marineau, Shigehiro Nomura, ShoGraphics Inc., Shunsuke Akiyama, Silicon Graphics Computer Systems Inc., Silicon Integrated Systems Corp Inc., Silicon Motion Inc., Simon P. Cooper, Snitily Graphics Consulting Services, Sony Corporation, Søren Sandmann, SRI, Stanislav Brabec, Stefan Dirsch, Stephan Dirsch, Stephan Lang, Steven Lang, Stuart Kreitman, Sun Microsystems Inc., SunSoft Inc., SuSE Inc, Sven Luther, Takis Psarogiannakopoulos, Takuma Murakami, Takuya SHIOZAKI, Tektronix Inc., The DOS-EMU-Development-Team, The Institute of Software Academia Sinica, The NetBSD Foundation, Theo de Raadt, Theodore Ts'o, The Open Group, The Open Software Foundation, The Regents of the University of California, The Santa Cruz Operation Inc., The Weather Channel Inc., The X Consortium, The XFree86 Project Inc., Thomas E. Dickey, Thomas G. Lane, Thomas Hellström, Thomas Mueller, Thomas Roell, Thomas Thanner, Thomas Winischhofer, Thomas Wolfram, Thorsten.Ohl, Tiago Gons, Todd C. Miller, Tomohiro KUBOTA, Torrey Lyons, Torrey T. Lyons, TOSHIBA Corp., Toshimitsu Tanaka, Travis Tilley, Tungsten Graphics Inc., Ty Sarna, UCHIYAMA Yasushi, Unicode Inc., UniSoft Group Limited, University of Utah, UNIX System Laboratories Inc., URW++ GmbH, VA Linux Systems, VIA Technologies Inc., Video Electronics Standard, VMware Inc., Vrije Universiteit, Wittawat Yamwong, Wyse Technology Inc., X Consortium, Xi Graphics Inc., X-Oz Technologies, X-TrueType Server Project and their contributors, Yu Shao,

This product includes software developed by The XFree86 Project, Inc (<http://www.xfree86.org/>) and its contributors.

This produce includes software that is based in part of the work of the FreeType Team (<http://www.freetype.org>).

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by Christopher G. Demetriou.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the X-Oz Technologies and its contributors.

CONTENTS

1.	Introduction to the X11R6.8.2 Release	1
2.	Summary of new features in X11R6.8.2	2
2.1	New X extensions	3
2.2	Distributed Multihead X (DMX)	4
2.3	Xprint updates	4
2.4	Updated Mesa and DRI from upstream sources	4
2.5	Video driver enhancements	4
3.	Drivers	6
3.1	Video Drivers	6
3.2	Print Drivers	7
3.3	Input Drivers	7
4.	Overview of X11R6.8.2	7
4.1	Loader and Modules	8
4.2	Configuration File	8
4.3	Command Line Options	8
4.4	XAA	8
4.5	Multi-head	8
4.6	Xinerama	9
4.7	DGA version 2	9
4.8	DDC	9
4.9	GLX and the Direct Rendering Infrastructure (DRI)	9
4.10	XVideo Extension (Xv)	10
4.11	X Rendering Extension (Render)	10
4.12	Other extensions	11
4.13	xedit	11
4.14	Font support	11
4.15	TrueType support	11
4.16	CID font support	11
4.17	Internationalisation of the scalable font backends	11
4.18	Large font optimisation	11
4.19	Unicode/ISO 10646 support	11
4.20	Xlib Compose file support and extensions	12
4.21	Bitstream Vera fonts	13
4.22	Luxi fonts from Bigelow and Holmes	13
5.	Miscellaneous	13
5.1	Legacy keyboard driver phase-out	13
5.2	Socket directory ownership and permissions	13
5.3	Composite exposes extra visuals	14
6.	Attributions/Acknowledgements/Credits	14