Release Notes for XFree86® 4.7.0

The XFree86 Project, Inc

August 2007

Abstract

This document contains information about the various features and their current status in the XFree86 4.7.0 release.

1. Introduction to the 4.x Release Series

XFree86 4.0 was the first official release of the XFree86 4 series. The current release (4.7.0) is the latest in that series. The XFree86 4.x series represents a significant redesign of the XFree86 X server, with a strong focus on modularity and configurability.

2. Configuration: a Quick Synopsis

Automatic configuration was introduced with XFree86 4.4.0 which makes it possible to start XFree86 without first creating a configuration file. This has been further improved in subsequent releases. If you experienced any problems with automatic configuration in a previous release, it is worth trying it again with this release.

While the initial automatic configuration support was originally targeted just for Linux and the FreeBSD variants, as of 4.5.0 it also includes Solaris, NetBSD and OpenBSD support. Full support for automatic configuration is planned for other platforms in future releases.

If you are running Linux, FreeBSD, NetBSD, OpenBSD, or Solaris, try Auto Configuration by running:

```
    XFree86 -autoconfig
```

If you want to customise some things afterwards, you can cut and paste the automatically generated configuration from the `/var/log/XFree86.0.log` file into an `XF86Config` file and make your customisations there. If you need to customise some parts of the configuration while leaving others to be automatically detected, you can combine a partial static configuration with the automatically detected one by running:

```
    XFree86 -appendauto
```

If you are using a platform that is not currently supported, then you must try one of the older methods for getting started like "`xf86cfg`", which is our graphical configuration tool. It also has a text mode interface that an be used to create an initial configuration file and customise existing configurations.

After that, you can use XFree86 server's ability to create a starting configuration file. To do so you must login as root, and run:

```
    XFree86 -configure
```

and then follow the instructions.

Finally, if all else fails, the trusty old standby text-based tool "`xf86config`" can also be used for generating X server config files.

Hopefully, at least one, and perhaps all of these various configuration options will give you a reasonable starting point for a suitable configuration file but we think that with the automatic mechanism you will probably not need any.

If you would like to customise the configuration file, see the XF86Config manual page. You should also check the driver-specific manual pages and the related documentation, which is found at *tables below* (section 4., page 6).

Before downloading any of the binary distributions for this release, read through the Installation Document as it can point out which particular binary you should download.

The next section describes what is **new** in the latest version (4.7.0) compared with the previous full release (4.6.0). There are many new features in this release and we unfortunately do not have enough space to cover them all here.

3. Summary of new features in 4.7.0.

This is a sampling of the new features in XFree86 4.7.0. A more complete list of changes can be found in the CHANGELOG that is part of the XFree86 source tree. It can also be viewed online at our CVSweb server `<URL:http://cvsweb.xfree86.org/cvsweb/xc/programs/Xserver/hw/xfree86/CHANGELOG?rev=HEAD>`.

3.1 Security Updates

A number of reported security issues have been addressed in this release. They include:

- CVE-2006-0747.
- CVE-2006-1861.
- CVE-2006-2661.
- CVE-2006-6101.
- CVE-2006-6102.
- CVE-2006-6103.
- CVE-2007-1003.
- CVE-2007-1351.
- CVE-2007-1352.
- CVE-2007-1667.
- CVE-2007-2754.

In addition, a number of potential integer overflow problems in the Type 1 font rendering code have been corrected.

3.2 Video Driver Enhancements

3.2.1 ati, atimisc, r128 and radeon

- Fix bug that caused the version check between the ati and atimisc modules to fail.

- The atimisc module's saving and restoring of the mode on server entry has been made more reliable on 64-bit systems.
- Fix bug that prevented detection of non-PCI adapters on systems that do not provide accurate PCI resource sizes.
- Fix the radeon driver's saving and restoring of the mode on server entry.

3.2.2 chips

- Fix some of this driver's endianness problems.

3.2.3 i128

- Resurrect support for Ti3026 RAMDACs found on the original I128 adapters. Note that there are still issues with depth 8 visuals and hardware cursors.

3.2.4 mga

- Add support for Matrox MGA G200e SE adapters.
- Fix determination of the mode on server entry. Also, save and restore it on VC switches.

3.2.5 newport

- Add support for the DPMS extension.

3.2.6 nv

- Add support for new hardware.

3.2.7 pnozz

- This accelerated driver supports the Weitek POWER 9100 SBUS adapters found in some SPARC laptops.

3.2.8 sunbw2, suncg14, suncg3, sunleo, suntcx

- These drivers, intended for various SPARC SBUS and UPA adapters, can now blank and unblank the screen correctly.

3.2.9 sunffb

- Prevent turning off the FPU on OpenBSD.
- Fix bug that prevented use of this driver on SunOS and Solaris.
- Workaround for VRAM corruption when unblanking FFB1 adapters.

3.2.10 trident

- Resave the mode on server entry when switching back to the server's virtual console.

3.3 Input Driver Enhancements

3.3.1 mouse

- Do not disable 3-button mouse emulation even when the mouse is found to actually have a third button. This helps laptops, and the like, running systems that combine a touchpad and an external mouse into one device.

3.4 XKB updates

- Add an XKB geometry for Dell Inspiron 8000 notebook keyboard.

- Add an XKB geometry for Enermax Aurora keyboards.
- Add XKB keycodes for MacIntosh USB keyboards.
- Add XKB definitions for X860x0 keyboards.
- Update Macintosh french keymap.
- Allow signed coordinates in XKB shape definitions.

3.5 X Server and Extension Updates

3.5.1 Loader

- Fixed support for `R_ALPHA_GPRELHIGH` relocations.
- Added support for `R_ALPHA_BRSGP` relocations.
- Fixed handling of archives on non-Linux SPARC.
- Fixed support for `R_SPARC_LO10` and `R_SPARC_OLO10` relocations.
- Added support for most ELF SPARC relocation types.
- Added an emulation of rand(3) calls.

3.5.2 XFree86 core server and modules

- Update -configure to detect when wscons support should be used for the keyboard on OpenBSD.
- Added preliminary support for PCI domains on OpenBSD/sparc64. This currently requires a kernel change.
- On the *BSD's, added support for absolute mouse position events and horizontal scroll wheels.
- Fixed a bug that caused an entire batch of mouse events to be rejected when one of an unknown type was encountered.
- Don't error out on unknown wskbd types on the *BSD's.
- Fixed memory leaks in option processing.
- Fixed area granularity handling in the framebuffer manager.
- Reworked the sizing of PCI resources. More specifically, reduced the X server's reliance on the OS to provide such information.
- Added support for SBUS and UPA adapters on NetBSD and OpenBSD.
- Added support for Solaris/amd64.
- Added support for UltraSPARC IV systems, including support for PCI Express.
- Added support for DragonFly, a variant of FreeBSD.
- Added support for x86_64 architectures on the *BSD's.
- Fixed page size determination in various server-related utilities.
- Made the mmapr and mmapw utilities understand wscons mapping modes on the *BSD's.

3.5.3 TinyX

- Added an option to the TinyX servers to prevent them from probing serial ports on Linux.

3.5.4 Xsun

- Changes to port the various Xsun servers to NetBSD/sparc have been merged in.

3.5.5 XDarwin

- This X server has been heavily upgraded by integrating a wide selection of changes from its now defunct X.Org implementation.

3.5.6 Xvfb

- Fixed segfaults in Xvfb's command line parsing.

3.5.7 cfb

- Fix a number of bugs in cfb24, even though it is no longer used by any driver provided in the distribution.
- Fix stipple handling on sparc64.

3.5.8 fb

- A problem that prevented the use of the GLX extension in some cases has been corrected.
- Fix fbCompose() for ABGR framebuffers.

3.5.9 Render

- Endianness fix for RenderSetPictureClipRectangles request.

3.6 Library, Client and Utility Updates

3.6.1 restest

- This XRes extension client, originally written by Mark Vojkovich, has been integrated, after rewriting it to prevent deadlocks that could occur when displaying its results in an X session.

3.6.2 xclock

- Make an analog xclock, when using Render, update the minute hand at most every minute instead of every update interval. This makes it consistent with non-Render mode.

3.6.3 xdm

- Fixed a number of memory leaks.
- Fixed bug that prevented an IPv6-aware xdm from accepting chooser connections.

3.6.4 xwd

- This utility has been extended to allow selection of a rectangular region to dump.

3.6.5 xterm

- Xterm patches upto and including #215 have been integrated.
- Change xterm, on various platforms, to not require that the OS define a utmp group.

3.7 I18n and Fonts

- Eliminated a number of issues related to the use of external fontconfig and freetype2 libraries.
- Updated the Serbian locale naming to reflect the country's name change from "Yugoslavia" to "Serbia and MonteNegro".

3.8 OS Support Updates

- Added support for PAM on Solaris and Darwin.
- Added support for OpenPAM on FreeBSD and NetBSD.

4. Drivers

4.1 Video Drivers

XFree86 4.7.0 includes the following video drivers:

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 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.

4.2 Input Drivers

XFree86 4.7.0 includes the following input drivers:

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

5. Known Problems

Currently no known problems are documented.

6. Overview of XFree86 4.x.

XFree86 4.x has a single X server binary called `XFree86`. 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.

XFree86 4.7.0 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. Support for additional architectures and operating systems is in progress and is planned for future releases.

6.1 Loader and Modules

The XFree86 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.

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.

Driver Name	Description	Further Information
`apm`	Alliance Pro Motion	README.apm
`ark`	Ark Logic	
`aspeed`	ASPEED Technology	
`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)
`pnozz` (+)	Weitek P9100	pnozz(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	
`vesa`	VESA	vesa(4)
`vga`	Generic VGA	vga(4)
`via`	VIA	via(4)
`vmware`	VMWare guest OS	vmware(4)
`xgi`	Xabre Graphics Inc	xgi(4)

Note about module security

The XFree86 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.

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	
eloinput	Elo 2500U USB	eloinput(4)
fpit	Fujitsu Stylistic Tablet PCs	fpit(4)
hyperpen	Aiptek HyperPen 6000	
js_x	JamStudio pentablet	js_x(4)
kbd	generic keyboards (alternate)	kbd(4)
keyboard	generic keyboards	keyboard(4)
magictouch	MagicTouch ProE-X	magictouch(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)

6.2 Configuration

The XFree86 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 XF86Config(5) manual page.

The XFree86 server has support for automatically determining an initial configuration on most platforms, as well as support or generating a basic initial configuration file.

6.3 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 XFree86(1) manual page.

6.4 XAA

The XFree86 Acceleration Architecture (XAA) was completely rewritten from scratch for XFree86 4.x. Most drivers implement acceleration by making use of the XAA module.

6.5 Multi-head

Some multi-head configurations are supported in XFree86 4.x, 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).

6.6 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 XFree86 4.x was completely rewritten for improved performance and correctness.

Known problems:

- Not all window managers are 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 XFree86 problem.

6.7 DGA version 2

DGA 2.0 is included in 4.7.0. 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.

6.8 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 will use this information to set the default monitor parameters and video mode when none are provided explicitly in the configuration file.

6.8.1 Changed behavior caused by DDC.

Several drivers use DDC information to set the screen size and pitch. This can be overridden by explicitly resetting it to the 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.

6.9 GLX and the Direct Rendering Infrastructure (DRI)

Direct rendered OpenGL® support is provided for several hardware platforms by the Direct Rendering Infrastructure (DRI), which is part of Mesa. Mesa also provides the 3D core rendering component of GLX. Further information can be found at the DRI Project's web site `<URL:http://dri.sf.net/>` and the Mesa web site `<URL:http://www.mesa3d.org>`.

6.10 XVideo Extension (Xv)

The XVideo extension is supported in XFree86 4.x. 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.

6.11 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.

XFree86 4.7.0 provides a partial implementation of Render sufficient for drawing anti-aliased text and image composition. Still to be implemented are geometric primitives and affine transformation of images.

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.

6.11.1 The Xft Library

On the client side, the Xft library provides access to fonts for applications using the FreeType library, version 2. FreeType currently supports Type1 and TrueType font files, a future release is expected to support BDF and PCF files as well, so Render applications will have access to the complete range of fonts available to core applications. 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 `XF86Config`.

To allow a graceful transition for applications moving from core text rendering to the Render extension, Xft can use either core fonts or FreeType and the Render extension for text. By default, Xft is configured to support both core fonts and FreeType fonts using the supplied version of FreeType 2. See the section on FreeType support in Xft for instructions on configuring XFree86 to use an existing FreeType installation.

The Xft library uses a configuration file, `XftConfig`, 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) man page.

6.11.2 FreeType support in Xft

XFree86 4.7.0 includes sources for FreeType version 2.1.8, and, by default, it is built and installed automatically.

6.11.3 Application Support For Anti-Aliased Text

Only three applications have been modified in this release to work with the Render extension and the Xft and FreeType libraries to provide anti-aliased text. Xterm, xditview and x11perf. 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.

6.12 Other extensions

The XFree86-Misc extension has not been fully ported to the new server architecture yet. This might 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.

6.13 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.

6.14 Font support

Details about the font support in XFree86 4.x can be found in the README.fonts document.

6.15 TrueType support

The XFree86 4.x server comes with a TrueType backend, known as the "FreeType" backend, based on the FreeType library. The functionality of the former X-TrueType was merged into the FreeType backend by the After X-TT Project for the XFree86 4.4.0 release, and the old X-TrueType backend has been dropped as of XFree86 4.5.0.

6.16 CID font support

Support for CID-keyed fonts is included in XFree86 4.x. The CID-keyed font format was designed by Adobe Systems `<URL:http://www.adobe.com>` for fonts with large character sets. The CID-keyed font support in XFree86 was donated by SGI `<URL:http://www.sgi.com>`. See the LICENSE document for a copy of the CID Font Code Public License.

6.17 Internationalisation of the scalable font backends

XFree86 4.x 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 their native encoding. This layer is used by the FreeType, Type1 and Speedo backends.

6.18 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.

6.19 Unicode/ISO 10646 support

What is included in 4.x:

- 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 font backend supports Unicode-encoded fonts.

6.20 Xlib Compose file support and extensions

A more flexible Compose file processing system was added to Xlib in XFree86 4.4.0. 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 "*<xlocaledir>*`/`*<localename>*`/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., "*<xlocaledir>*`/`*<localename>*`/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.

6.21 Luxi fonts from Bigelow and Holmes

XFree86 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>`.

7. Credits

This section lists the credits for the XFree86 4.7.0 release. For a more detailed breakdown, refer to the CHANGELOG file in the XFree86 source tree, the cvs-commit archives `<URL:http://www.mail-archive.com/cvs-commit@xfree86.org/maillist.html>`, or the ’cvs log’ information for individual source files.

New Features, Enhancements and Updates:

Security Updates:
Josh Bressens, Chris Evans, Matthieu Herrb, Marc La France, Sean Larsson, Victor Stinner, iDefense.

NVIDIA driver updates and new hardware support:
Mark Vojkovich

Improved SPARC support:
Marc La France.

Xterm enhancements and updates:
Thomas Dickey.

DragonFly support:
David H. Dawes.

Integration:

General Integration of Submissions:
Marc La France, David H. Dawes.

Release Engineering:
David H. Dawes.

Patches and other submissions (in alphabetical order):
Andrew Aitchison, Marc Balmer, Etienne Bersac, Martin Bochnig, Peter Breitenlohner, Josh Bressers, James Chacon, Yves de Champlain, Alan Coopersmith, David Dawes, Eike Dehling, Thomas Dickey, Matthias Drochner, Jay Estabrook, Chris Evans, Will L G, Andriy Gapon, Charles M. Hannum, Frank J. R. Hanstick, Ben Harris, Matthieu Herrb, Iain Hibbert, Martin Husemann, iDefense, Milos Komarcevic, Marc La France, Sean Larrson, Michael Lorenz, Jie Luo, Loic Mahe, Minoura

Makoto, Martin Mares, Luke Mewburn, NetBSD, Dmitry Pervushin, Alexander Pohoyda, Ty Sarna, SciFi, Christopher Sekiya, Jamey Sharp, Victor Stinner, Frank van der Linden, Shin Takemura, Mark Vojkovich, Nathan J. Williams, X.Org, Christos Zoulos.

Webmaster:
Georgina O. Economou

Hosting:
Internet Systems Consortium, Inc `<URL:http://www.isc.org/>`.

8. Attributions/Acknowledgements

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

This product 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 X-Oz Technologies (http://www.x-oz.com/).

CONTENTS

`$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/RELNOTES.sgml,v 1.143 2007/08/04 02:16:08 tsi Exp $`