Release Notes for XFree86™ 4.3.0

The XFree86 Project, Inc

26 February 2003

Abstract

This document contains some information about features present in XFree86 4.3.0 and their status.

1. Introduction to the 4.x Release Series

XFree86 4.0 was the first official release of the new XFree86 4 series. The current release (4.3.0) is the latest in that series. XFree86 4 represents a significant redesign of the XFree86 X server. Not all of the hardware drivers from 3.3.x have been ported to 4.x yet, but conversely, 4.x has support for a lot of hardware that is not supported in 3.3.x. Our Driver Status document summarizes how the hardware driver support compares between 3.3.6 and 4.3.0. Please check there first before downloading 4.3.0.

The 4.0.1 release introduced a new graphical configuration tool, "`xf86cfg`", and a text mode interface was added to it for the 4.0.2 release. It is the preferred configuration tool provided by with XFree86. The trusty old text-based tool "`xf86config`" can also be used for generating X server config files. In addition to these tools, the XFree86 server has some built in capabilities for generating a base config file. This works well for most hardware, and in most cases is the easiest way to get an initial config file. To try it out, just run (as root):

```
XFree86 -configure
```

Each of these configuration options will give you a reasonable starting point for a suitable configuration file. We've put some effort into documenting the 4.3.0 config file format, and you can find that information in the XF86Config manual page. Check there and the driver-specific manual pages and the related documentation for further information. References to this driver-specific information can be found in the *tables below* (section 3., page 4).

We have plans to make the configuration file optional in a future release. The XFree86 server is close to being able to automatically determine a complete base configuration for most popular hardware configurations.

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

The next section describes what is new in the latest version (4.3.0) compared with the previous full release (4.2.0). The other sections below describe some of the new features and changes between 3.3.x and 4.0. There are lot's of new features, and we definitely don't have enough space to cover them all here.

2. Summary of new features in 4.3.0.

2.1 Video Driver Enhancements

- ATI Radeon 9x00 2D support added, and 3D support added for the Radeon 8500, 9000, 9100, and M9. The 3D support for the Radeon now includes hardware TCL.
- Support added to the i810 driver for Intel 845G, 852GM, 855GM and 865G integrated graphics chipsets, including 2D, 3D (DRI) and XVideo. Support for the 830M has been improved, and XVideo support added.
- National Semiconductor SC1x00, GX1, and GX2 chipset support added with the "nsc" driver.
- Support added for the NVIDIA nForce2 integrated graphics, GeForce 4, and GeForce FX.
- Major SiS driver updates for some of the latest chipsets. Unfortunately the SiS 3D driver has had to be disabled because no one has yet taken up the challenge to port it to Mesa 4.x.
- The s3virge driver now has support for double scan modes on the DX (with XVideo disabled).
- Updates to the savage driver, including fixing problems with the TwisterK, and problems with incorrect memory size detection.
- 2D acceleration added for the Trident CyberBladeXP/Ai1 chipsets.
- Support for big endian architectures has been added to the C&T driver.
- Various updates and bug fixes have been made to most other drivers.

2.2 Input Driver Enhancements

- The mouse driver now has automatic protocol detection for PS/2 mice.
- Several new input drivers have been added, including tek4957, jamstudio (js_x), fpit, palmax, and ur98 (Linux only).

2.3 X Server and Extension Updates

- Support for the RandR extension has been partially integrated into the XFree86 server, providing support for resizing the root window at run-time.
- The Mesa version used for OpenGL® 1.3 and DRI driver support has been updated to 4.0.4.
- The XFree86 server's hot keys (including those for switching modes and virtual terminals) can now be configured via XKB. Previously they were hard coded. An X server configuration option has been added to allow the VT switching hot keys to be disabled.

2.4 Client and Library Updates

- An Xcursor library providing support for alpha blended (ARGB) and animated cursors. Two Xcursor themes are provided (redglass and whiteglass), as well as the default "core" theme (the traditional cursors).
- Xterm updated to patch level 173, including the following bugfixes:
  - Fix two infinite loops (special cases of mouse hilite tracking, DECUDK parsing).
  - Make repainting of the 256-color example work properly.
  - Modify parser tables to improve detection of malformed control sequences, making xterm behave more like a real DEC terminal.

- Fix a problem with the blinking cursor which occasionally caused xterm to pause until a key was pressed.
- Fix improper parsing of multiple items in the ttyModes resource.

and the following improvements:

- Modify xterm to invoke luit.
- Add simple session management client capabilities.
- Add a modifyCursorKeys resource to control how the shift- and similar modifiers are used to make a cursor escape sequence.
- Check if the printerCommand resource string is empty, and use this to allow the user to disable printer function.
- Sort the options list which is displayed in help- and syntax-messages at runtime to simplify maintenance.

2.5 I18N and Font Updates

- FreeType2 updated to version 2.1.1.
- The "freetype" X server font backend has undergone a partial rewrite. The new version is based on FreeType 2, and handles TrueType (including OpenType/TTF), OpenType/CFF and Type 1 fonts. The old "type1" backend is now deprecated, and is only used for CIDFonts by default.
- A new utility called "mkfontscale", which builds fonts.scale files, has been added.
- The Xft library has undergone a major restructuring, and is now split into fontconfig (which deals with font discovery and configuration and is independent from X), and Xft itself (which uses fontconfig and deals with font rasterisation and rendering. The format of the Xft font configuration files has changed in an incompatible manner.
- Support has been added to the Xft library to do rendering with the core X11 protocol. This allows clients using this library to render to X servers that don't have support for the RENDER extension.
- There has been a significant reworking of the XKB support to allow multi-layout configurations. Multi-layout configurations provide a flexible way of supporting multiple language layouts and switching between them.

2.6 OS Support Updates

- Updates for Darwin/Mac OS X, including:
  - Indirect GLX acceleration added.
  - Smaller memory footprint and faster 2-D drawing in rootless mode.
  - Full screen mode now uses shadowfb for much faster 2-D drawing.
  - Native fonts can be used on MacOS X.
- Various Cygwin support updates, including an experimental rootless X server for Cygwin/XFree86.
- AMD x86-64 support (primarily for Linux so far) has been added.

- Support added for OpenBSD/sparc64.
- Major OS/2 support updates.
- Major SCO OpenServer updates.
- Multi-head support has been added for 460GX-based Itanium systems, and for ZX1-based Itanium2 systems.
- Experimental support for SunOS/Solaris on UltraSPARC systems.

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. Drivers

3.1 Video Drivers

XFree86 4.3.0 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 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.

XFree86 4.3.0 includes the following input drivers:

3.2 Input Drivers

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

4. Overview of XFree86 4.x.

Unlike XFree86 3.3.x where there are multiple X server binaries, each of which drive different hardware, XFree86 4.3.0 has a single X server binary called `XFree86`. This binary can either have one or more video drivers linked in statically, or, more usually, dynamically load the video drivers and other modules that are needed.

XFree86 4.3.0 has X server support for most UNIX(R) and UNIX-like operating systems on Intel/x86 platforms, plus support for Linux on Alpha, PowerPC, IA-64, Sparc, and Mips platforms, and for Darwin on PowerPC. Work on support for additional architectures and operating systems is in progress, and is planned for future releases.

4.1 Loader and Modules

The XFree86 X server has a built-in run-time loader, donated by Metro Link <URL:http://www.metrolink.com>. This loader can load normal object files and libraries in most of the commonly used formats. Since the loader doesn't rely on an operating system's native dynamic loader support, it works on platforms that don't provide this feature, and makes it possible for the modules to be operating system independent (although not, of course, independent of CPU architecture). This 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.

One of the main benefits of this loader is that when modules are updated, they do not need to be

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	
vesa	VESA	vesa(4)
vga	Generic VGA	vga(4)
vmware	VMWare guest OS	vmware(4)

recompiled for every different operating system. In the future we plan to take advantage of this to provide more frequent driver module updates in between major releases.

The loader in version 4.3.0 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 (API and ABI) used in this release is still 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.

Note about module security

Driver Name	Description	Further Information
calcomp	Calcomp	
citron	Citron	citron(4)
digitaledge	DigitalEdge	
dmc	DMC	dmc(4)
dynapro	Dynapro	
elographics	EloGraphics	
elographics	EloGraphics	
fpit	Fujitsu Stylistic Tablet PCs	fpit(4)
hyperpen	HyperPen	
js_x	JamStudio pentablet	js_x(4)
kbd	generic keyboards (alternate)	kbd(4)
keyboard	generic keyboards	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)

```
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 intro-
duction of viruses and contaminated code can occur and wreak
havoc on your system. We hope to have a mechanism for sign-
ing/verifying the modules that we provide available in a future
release.
```

4.2 Configuration File

The X server configuration file format has been extended to handle some of the new functionality. The `xf86config` utility can be used to generate a basic config file, that may require some manual editing. The X server also has preliminary support for generating a basic config file. This is done by running (as root) "`XFree86 -configure`". Alternatively, the sample config file `XF86Config.eg` that is installed in `/usr/X11R6/lib/X11` may be used as a starting point. The `xf86cfg` utility can also be used to create a new configuration file or to edit an existing one. The `XF86Setup` utility is currently not usable, but work is continuing in this area.

The main changes are covered here, but please refer to the XF86Config(5) manual page for more comprehensive information:

- The Module section is used to load server extension modules and font modules, but not XInput drivers. The `.so` suffix should no longer be specified with module names. Options may be supplied for modules by loading the module via a SubSection instead of the usual Load keyword. The `bitmap` module is the only font module that is loaded by default. No server extensions are loaded by default, but some are built-in to the server. It is strongly recommended that the extension module containing a range of small miscellaneous extensions (`extmod`) be loaded because some commonly used things won't work correctly without it. The following example shows how to load all the server extensions plus the Type1 and TrueType fonts support, and a commented example that shows how to pass options to an extension (this one is for loading the misc extensions (`extmod`) with the XFree86-VidModeExtension disabled):

```
Section "Module"

    Load "dbe"
    Load "record"
    Load "glx"
    Load "extmod"

    Load "type1"
    Load "freetype"

  # SubSection "extmod"
  #     Option "Omit XFree86-VidModeExtension"
  # EndSubSection

EndSection
```

- Option flags have been extended and are now used more widely in the config file. Options flags come in two main types. The first type is exactly like the old form:

```
    Option "name"
```

  where the option just has a name specified. The name is case insensitive, and white space and underscore characters are ignored. The second type consists of a name and a value:

```
    Option "name" "value"
```

  The value is passed transparently as a string to the code that uses the option. Common value formats are integer, boolean, real, string and frequency. The following boolean option values are recognised as meaning TRUE: `"true"`, `"yes"`, `"on"`, `"1"`, and no value. The values recognised as FALSE are `"false"`, `"no"`, `"off"`, `"0"`. In addition to this, `"no"` may be prepended to the *name* of a boolean option to indicate that it is false. Frequency options can have the strings `Hz`, `kHz`, or `MHz` appended to the numerical value specified.

  Note: the value must always be enclosed in double quotes (`"`), even when it is numerical.

- The ServerFlags section now accepts its parameters as Options instead of as special keywords. The older keyword format is still recognised for compatibility purposes, but is deprecated and support for it will likely be dropped in a future release. The DPMS and screen save timeout values are now specified in the ServerFlags section rather than elsewhere (because they are global parameters, not screen-specific). This example shows the defaults for these:

```
        Option "blank time"    "10"
        Option "standby time"  "20"
        Option "suspend time"  "30"
        Option "off time"      "40"
```

The new option `AllowDeactivateGrabs` allows deactivating any active grab with the key sequence `Ctrl+Alt+Keypad-Divide` and the new option `AllowClosedownGrabs` allows closing the connection to the grabbing client with the key sequence `Ctrl+Alt+Keypad-Multiply`. Note that these options are off by default as they allow users to remove the grab used by screen saver/locker programs.

- The Keyboard, Pointer and XInput sections have been replaced by a more general InputDevice section. The old Keyboard and Pointer sections are still recognised for compatibility purposes, but they are discommended and support for them may be dropped in future releases. The old XInput sections are no longer recognised. The keywords from the old sections are expressed as Options in the InputDevice sections. The following example shows typical InputDevice sections for the core mouse and keyboard.

```
Section "InputDevice"
    Identifier  "Keyboard 1"
    Driver      "keyboard"
    Option      "AutoRepeat" "500 5"
    Option      "XkbModel"   "pc104"
    Option      "XkbLayout"  "us"
EndSection

Section "InputDevice"
    Identifier  "Mouse 1"
    Driver      "mouse"
    Option      "Protocol"   "PS/2"
    Option      "Device"     "/dev/mouse"
    Option     "SampleRate" "80"
EndSection
```

- The Monitor section is mostly unchanged. The main difference is that a set of VESA modes is defined internally in the server, and so for most monitors, it isn't necessary to specify any modes explicitly in the Monitor section. There is also a new Modes section that can be used to define a set of modes separately from the Monitor section, and the Monitor section may "include" them with the "`UseModes`" keyword. The Monitor section may also include Options. Options that are monitor-specific, like the `"DPMS"` and `"Sync on Green"` options are best specified in the Monitor sections.
- The Device sections are mostly unchanged. The main difference is the new (and mandatory) Driver keyword that specifies which video driver should be loaded to drive the video card. Another difference is the BusID keyword that is used to specify which of possibly multiple video cards the Device section is for. The following is an example for a Matrox card:

```
Section "Device"
    Identifier "MGA 1"
    Driver     "mga"
    BusID      "PCI:1:0:0"
EndSection
```

- The Screen sections are mostly unchanged. The old Driver keyword is no longer used, and a mandatory Identifier keyword has been added. The DefaultColorDepth keyword has been renamed to DefaultDepth.
- A new section called ServerLayout has been added to allow the layout of the screens and the selection of input devices to be specified. The ServerLayout sections may also include options that are normally found in the ServerFlags section. Multiple ServerLayout sections may be present, and selected from the command line. The following example shows a ServerLayout section for a dual-headed configuration with two Matrox cards, and two mice:

```
Section "ServerLayout"
    Identifier  "Layout 1"
    Screen      "MGA 1"
    Screen      "MGA 2" RightOf "MGA 1"
    InputDevice "Keyboard 1" "CoreKeyboard"
    InputDevice "Mouse 1"    "CorePointer"
    InputDevice "Mouse 2"    "SendCoreEvents"
    Option      "BlankTime"  "5"
EndSection
```

  See the XF86Config(5) man page for a more detailed explanation of the format of the new ServerLayout section.

The config file search patch has been extended, with the directories `/etc/X11` and `/usr/X11R6/etc/X11` being added. The full search path details are documented in the XF86Config manual page.

4.3 Command Line Options

The following new X server command line options have been added:

−depth *n*

This specifies the colour depth that the server is running at. The default is 8 for most drivers. Most drivers support the values 8, 15, 16 and 24. Some drivers also support the values 1 and 4. Some drivers may also support other depths. Note that the depth is different from the ‘‘bpp’’ that was specified with previous versions. The depth is the number of bits in each pixel that are significant in determining the pixel’s value. The bpp is the total size occupied by each pixel, including bits that are not used. The old −bpp option is no longer recognised because it isn’t a good way of specifying the server behaviour.

−fbbpp *n*

This specifies the bpp format to use for the framebuffer. This may be used in 24-bit mode to force a framebuffer format that is different from what the driver chooses by default. In most cases there should be no need to use this option.

−pixmap24

This specifies that the client-side pixmap format should be the packed 24-bit format that was often used by the 3.3.x servers. The default is the more common 32-bit format. There should normally be no need to use this option.

−pixmap32

This specifies that the client-side pixmap format should be the sparse 32-bit format. This is the default, so there should normally be no need to use this option.

−layout *name*

This specifies which ServerLayout section in the config file to use. When this option is not specified, the first ServerLayout section is used. When there is no ServerLayout section, the first Screen section is used.

−screen *name*

This specifies which Screen section in the config file to use. When this option is not specified, the first ServerLayout section is used. When there is no ServerLayout section, the first Screen section is used.

−keyboard *name*

This specifies which InputDevice section in the config file to use for the core keyboard. This option may be used in conjunction with the −screen option.

-pointer *name*

This specifies which InputDevice section in the config file to use for the core pointer. This option may be used in conjunction with the -screen option.

-modulepath *path*

This specifies the module search path. The path should be a comma-separated list of absolute directory paths to search for server modules. When specified here, it overrides the value specified in the config file. This option is only available when the server is started by the root user.

-logfile *file*

This specifies the log file name. When specified here, it overrides the default value. This option is only available when the server is started by the root user.

-scanpci

This specifies that the scanpci module should be loaded and executed. This does a scan of the PCI bus.

-logverbose [*n*]

This options specifies the verbosity level to use for the log file. The default is 3.

The following X server command line options have been changed since 3.3.x:

-verbose [*n*]

This option specifies the verbosity level to use for the server messages that get written to stderr. It may be specified multiple times to increase the verbosity level (as with 3.3.x), or the verbosity level may be specified explicitly as a number. The default verbosity level is 0.

-xf86config *filename*

This option has been extended to allow non-root users to specify a relative config file name. The config file search path will be used to locate the file in this case. This makes it possible for users to choose from multiple config files that the the sysadmin has provided.

A more complete list of XFree86 X server command line options can be found in the XFree86(1) manual page.

4.4 XAA

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

4.5 Multi-head

Some multi-head configurations are supported in XFree86 4.x, primarily with multiple PCI/AGP cards. However, this is an area that is still being worked on, and we expect that the range of configurations for which it works well will increase in future releases. A configuration that is known to work well in most cases is multiple (supported) Matrox 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 XFree86 4.x 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 XFree86 problem.

4.7 DGA version 2

DGA 2.0 is included in 4.3.0, but is not implemented by all drivers. Preliminary documentation for the client libraries can be found in the `README.DGA` document. A good degree of backwards compatibility with version 1.0 is provided.

4.8 DDC

The VESA(R) 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 or warn about monitor sync limits (but not modelines yet). For some drivers, the X server's new `-configure` option uses the DDC information when generating the config file.

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)

Precision Insight <URL:http://www.precisioninsight.com> was provided with funding and support from Red Hat <URL:http://www.redhat.com>, SGI <URL:http://www.sgi.com>, 3Dfx <URL:http://www.3dfx.com>, Intel <URL:http://www.intel.com>, ATI <URL:http://www.ati.com>, and Matrox <URL:http://www.matrox.com> to integrate the GLX extension for 3D rendering in an X11 window. The 3D core rendering component is the Mesa <URL:http://www.mesa3d.org> library. SGI has released the sources to the GLX extension framework under an open license, which essentially provides the glue between the 3D library and this windowing system. Precision Insight has integrated these components into the XFree86 X Server and added a Direct Rendering Infrastructure (DRI). Direct Rendering provides a highly optimized path for sending 3D data directly to the graphics hardware. This release provides a complete implementation of direct rendering support for the 3Dfx Banshee, Voodoo3 and Voodoo5 graphics cards, as well as the Intel i810/i815 cards, ATI Rage 128, and Matrox G400. Updated information on DRI compatible drivers can be found at the DRI Project <URL:http://dri.sourceforge.net> on SourceForge <URL:http://www.sourceforge.net>.

4.10 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.

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.

XFree86 4.3.0 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.

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. 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.

4.11.2 FreeType support in Xft

XFree86 4.3.0 includes sources for FreeType version 2.1.1, and, by default, they are built and installed automatically.

4.11.3 Application Support For Anti-Aliased Text

Only three applications have been modified in XFree86 4.3.0 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.

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.

4.14 Font support

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

4.15 TrueType support

XFree86 4.x comes with two TrueType backends, known as ‘xfsft’ (the "freetype" module) and ‘X-TrueType’ (the "xtt" module). Both of these backends are based on the FreeType library.

4.16 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.

4.17 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 uses fonts in encodings other than their their native encoding. This layer is used by the Type1 and Speedo backends and the ‘xfsft’ version of the TrueType backend. The ‘X-TrueType’ version of the TrueType backend uses a different re-encoding method based on loadable encoding modules.

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 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).
- Both the xfsft (the "freetype" module) and the X-TrueType (the "xtt" module) TrueType font backends support Unicode-encoded fonts.

4.20 Luxi fonts from Bigelow and Holmes

XFree86 now 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>`.

4.21 Directory rearrangements

Some changes to the installed XFree86 directory structure have been implemented for 4.x. One important change is a modified search path for the X server's `XF86Config` file. The details of this can be found in the XF86Config manual page. The other main change is moving most of the run-time configuration files to `/etc/X11`, with symbolic links in the old `/usr/X11R6/lib/X11` location pointing to the new location. Some run-time generated files are now located under the appropriate subdirectories of `/var`, again with the relevant symbolic links in the old location.

CONTENTS

$XFree86: xc/programs/Xserver/hw/xfree86/doc/sgml/RELNOTES.sgml,v 1.81 2003/02/27 00:45:05 dawes