Release Notes for XFree86™ 4.0.2

The XFree86 Project, Inc

17 December 2000

Abstract

This document contains some information about the features present in XFree86 4.0.2 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 (4.0.2) 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 some hardware support not present in 3.3.x. Our Driver Status document summarizes how the hardware driver support compares between 3.3.6 and 4.0.2. Please check there first before downloading 4.0.2.

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 work in progress, but definitely worth trying out. The trusty old text-based tool "`xf86config`" can also be used for generating X server config files. In addition to these tools, we've been working on a configuration tool that is built-in to the X server. It is included in the release, and it works well for some hardware. 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 4.0.2 config file format, and you can find that information in the XF86Config manual page. Check that, the driver manual pages and the related documentation for further information.

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 (4.0.2). 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 4.0.2.

2.1 X server

- Darwin/Mac OS X is now supported and the X server runs on PowerPC. The X server for this platform is called "Xdarwin", and its installation and configuration is different from the "XFree86" server used on other platforms. Please read the XFree86 on Darwin and Mac OS X document for further information.

- Mesa has been updated to the 3.4 stable release.
- A driver for ATI Radeon adapters has been added.
- ATI driver support for multi-head configurations and non-Intel platforms has been improved.
- The ATI driver has been changed to invoke the appropriate driver for any Rage 128 and Radeon adapters that it finds in the system.
- A driver (i128) for Number Nine chipsets has been added.
- A driver (savage) for S3 Savage chipsets has been added.
- A driver (siliconmotion) for some Silicon Motion chipsets has been added.
- The driver (ark) for Ark Logic chips has been ported to 4.x.
- A VESA driver has been added.

- A driver for SGI newport cards has been added (Linux/mips).
- The trident driver has lots of fixes to the CyberBlade/Blade support, XVideo support for the Image/Blade series (although scaling doesn't work yet on the Image series).
- The s3virge driver has stabilized ViRGE GX2 support, includes ViRGE DX and ViRGE support for XVideo YUV images, and various fixes.
- The 3Dlabs (glint) driver now has Permedia3 support.
- The SiS driver has had many updates, and XVideo support for the 630 is included.
- The NVIDIA (nv) driver has been updated to include support for the GeForce2, and line acceleration has been added. Also, DDC support has been improved, and support added for Alpha platforms (dense only).
- The neomagic driver has had various bug fixes and extended acceleration support. DGA support has also been added.
- The Chips and Technologies driver (chips) has initial support for the 69030.
- The tseng driver has multi-head fixes and DGA support has been added.
- Most video drivers have been converted to use the integrated "fb" framebuffer code instead of the old "cfb" code.
- DRI support has been updated for the Rage 128, 3Dfx Voodoo3 and Voodoo5, Intel i810/i815 and Matrox G400. DRI support for SiS and Sun Creator3D is available. The DRI is now also available on Alpha platforms.
- An input driver (citron) for Citron Infrared Touch devices has been added.
- An input driver (penmount) for PenMount devices has been added.
- An input driver (digitaledge) for DigitalEdge devices has been added.
- Big endian problems in Xinerama have been fixed.
- The new "render" extension has been added (see *below* (section 4.11, page 13)). The design has been extended in the following ways:
  - A new FillRectangles request was added.
  - The 'mask' operand was changed to allow per-channel alphas.

  Server implementation:

  - Implemented remaining code for client clip lists.
  - Fixed plenty of bugs related to text rendering.
  - Complete compositing code, except for separate alpha.

  Todo:

  - Polygons
  - Image scaling
  - Separate alpha channels
  - Support for visuals other than TrueColor.

  Drivers with render extension support include: i128, glint, ati (r128 and radeon), chips, cirrus, mga, neomagic, nv, tdfx, trident, tseng, vesa, s3virge.

  Acceleration for the render extension:

  - XAA infrastructure for acceleration.
  - Experimental MGA acceleration using XAA
  - kdrive (TinyX) infrastructure
  - Experimental TinyX Trident 9525DVD acceleration

  Xft library:

  - New font naming/access library.
  - Abstraction for core/Render text.
  - Uses FreeType2 rasterizer.
  - Allows anti-aliased/subpixel sampled text.
  - Gracefully falls back to core rendering.
  - Complicated font matching mechanism.
  - Uses server DPI to convert point sizes to pixel sizes, even for core fonts.

  xterm:

  - Can use Xft, by default uses core fonts.
  - Uses new options '`-fa`' for family name and '`-fs`' for font size (in points).

  x11perf:

  - Add tests for anti-aliased, subpixel sampled and aliased fonts using the Xft library.

  xditview:

  - When the Xft library is build, uses Xft for all font access.

  Qt, Gtk, twm:

  - Qt changes available here `<URL:http://XFree86.org/~keithp/download/qtkernel.tar.bz2>`.
  - Gtk changes in process.
  - twm hacks should never see the light of day.

- `xf86cfg`, a new graphical configuration tool for XFree86 4.x, and can be used to either write the initial configuration or make customisations to the current configuration. `xf86cfg` is a work in progress, and allows configuration of:

- Specific setup of monitors, cards, keyboards and mice, as well as adding or removing them.
- Server layout setup, allowing complex configuration of physical monitor positions, default color depth and/or rotated monitors.
- Mode line editor that can be used to configure multiple monitors, and allows adding a modeline specific to a monitor directly to the configuration file.
- AccessX configuration interface, that provides an interface to easy setup of most AccessX options, including:
  - Timeout to reset controls.
  - StickyKeys, for people with disabilities that cannot press two keys at the same time.
  - MouseKeys, mouse control only with the keyboard.
  - RepeatKeys, repeat rate and repeat delay.
  - SlowKeys, to avoid pressing keys accidentally, they're only accepted if pressed for some specific amount of time.
  - BounceKeys, helps avoiding multiple key presses by only accepting a key if it is pressed only once and not pressed again in a specific amount of time.

New configuration options are being worked on, as well as correcting some of the bugs in the current options. A protocol for plugging in external modules is also planned.

2.2 X libraries and clients.

- Significant updates to the internationalisation support in Xlib (see *below* (section 2.3, page 4)).
- Some xfs updates, including font path verification, and new options.
- XTerm updates.
- New "xvinfo" client for querying the XVideo extension.

2.3 Fonts and Internationalisation

- Many of the "misc" bdf fonts have been updated and extended, and a wider range of ISO-8859 subsets have been added. These are now auto-generated from ISO-10646 encoded master fonts.
- The ClearlyU ISO-10646 encoded fonts have been updated.
- Functions to read keyboard input in locale independent UTF-8 encoding have been added to libX11: Xutf8LookupString, Xutf8ResetIC.
- Functions to output strings in locale independent UTF-8 encoding have been added to libX11: Xutf8DrawString, Xutf8DrawImageString, Xutf8TextEscapement, Xutf8TextExtents, Xutf8TextPerCharExtents, Xutf8DrawText.
- Functions to convert between Compound Text or locale dependent encoding and UTF-8 have been added to libX11: Xutf8TextListToTextProperty, Xutf8TextPropertyToTextList. The converter between Compound Text and UTF-8 in Xlib has been improved; a round-trip conversion now correctly converts all graphic Unicode characters back and forth.
- libXaw now offers selected text using both selection targets, UTF8_STRING and COMPOUND_TEXT.

- Locales with UTF-8 encodings are now supported; but the UTF-8 displaying facilities are not adequate yet.
- XKB keyboard definitions have been added and updated for some countries.

2.4 Platforms

- Darwin/Mac OS X.
- Greatly improved IA-64 support.
- Improved Linux/mips support.
- Support has been added for more Alpha platforms under Linux. This now includes all platforms that require sparse memory mapping.

3. Drivers

3.1 Video Drivers

XFree86 4.0.2 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 4.0.2 includes the following input drivers:

3.2 Input Drivers

4. Summary of XFree86 4.

Unlike XFree86 3.3.x where there are multiple X server binaries, each of which drive different hardware, XFree86 4.0.2 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 4.0.2 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 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.

Driver Name	Description	Further Information
apm	Alliance Pro Motion	README.apm
ark	Ark Logic	
ati	ATI	README.ati, README.r128, r128(4)
chips	Chips & Technologies	README.chips, chips(4)
cirrus	Cirrus Logic	
cyrix (*)	Cyrix MediaGX	README.cyrix
fbdev	Linux fbdev	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 i810	README.i810, i810(4)
imstt	Integrated Micro Solns	
mga	Matrox	mga(4)
neomagic	NeoMagic	neomagic(4)
newport (-)	SGI Newport	README.newport, newport(4)
nv	NVIDIA	nv(4)
rendition	Rendition	README.rendition, rendition(4)
s3virge	S3 ViRGE	README.s3virge, s3virge(4)
savage	S3 Savage	savage(4)
siliconmotion	Silicon Motion	siliconmotion(4)
sis	SiS	README.SiS
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	
tga	DEC TGA	README.DECtga
trident	Trident	trident(4)
tseng	Tseng Labs	
vesa	VESA	vesa(4)
vga	Generic VGA	vga(4)

Driver Name	Description	Further Information
acecad	AceCad	
citron	Citron	citron(4)
digitaledge	DigitalEdge	
dynapro	Dynapro	
elographics	EloGraphics	
keyboard	generic keyboards	keyboard(4)
microtouch	MicroTouch	
mouse	most mouse devices	mouse(4)
mutouch	MicroTouch	
penmount	PenMount	
spaceorb	SpaceOrb	
summa	SummaGraphics	
void	dummy device	void(4)
wacom	Wacom tablets	wacom(4)

The loader in version 4.0.2 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 (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

> 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.

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 `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 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 "pex5"
    Load "xie"
    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 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 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 1.

`-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.

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 4.0.2, 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, but it does not yet use it the determine modelines. For some drivers, the X server's new `-configure` option uses the DDC information when generating the config file.

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> (now part of the Professional Services group at VA Linux Systems <URL:http://www.valinux.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 4.0.2 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 only core fonts, see the section on building FreeType support for the changes needed to add FreeType/Render fonts.

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 man page.

4.11.2 Building FreeType support for Xft

XFree86 4.0.2 includes sources for FreeType version 2.0.1, but they are not built and installed automatically. As a result, Xft is configured to provide only core fonts by default.

To build FreeType support for Xft, first FreeType must be built and installed, either from the sources included in XFree86 in `extras/freetype2` or from another FreeType (version 2.0.1 or later) release. Early FreeType version 2 releases used a different header file installation and aren't compatible with XFree86. Instructions for building and installing FreeType can be found in the `INSTALL` file included with the FreeType release.

Second, XFree86 needs to know the installed location for FreeType, usually `/usr/local`. Edit (or create) `config/cf/host.def` to include:

```
#define Freetype2Dir    /usr/local
```

Finally, build XFree86 with "make World" from the top.

4.11.3 Application Support For Anti-Aliased Text

Only three applications have been modified in XFree86 4.0.2 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 two command line options and associated resources to direct it to use Xft instead:

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

- `-fs` pointsize / `.VT100.faceSize:` pointsize. Selects the pointsize.

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 has not been optimized within the X server yet.

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 new `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 Xaw

Two versions of the Xaw library are provided with XFree86 4.x. A version with bug fixes and a few binary compatible improvements and a new version with several new features.

New features:

- A `displayList` resource is available to all Xaw widgets. It basically consists of a list of drawing commands, fully described in the `Xaw(3)` manual page, that enables a integration of Xaw programs with the new window/desktop managers that allows for configurable themes.
- Some new actions were added to all Xaw widgets, to allow more configurable control of the widgets, and to allow setting resources at run time.
- Since Xpm was integrated into XFree86, programs linked with the new Xaw library will also link with Xpm. This allows for color background pixmaps, and also for shaped widgets.
- The text widget is the widget that will present more changes. These include:
  - Block cursor.
  - Compile time limit of 16384 undo/redo levels (that will automatically grow if the text is not saved when this mark is reached).
  - Overwrite mode.
  - Text killed is inserted in a kill ring list, this text is not forgotten, pressing `M-y` allows traversing the kill ring list.
  - International support for latin languages is available even if the `international` resource is not set. Users will need to properly set the *locale* environment to make complete use of this feature.
  - A better `multiply` interface is provided. Pressing `C-u,<number>` (where number can be negative) allows passing parameters for text actions.
  - Text can be formatted to have left, right, center or full justification.
  - Text indentation support is also available.

Bug fixes:

- The simple menu widget geometry management code was improved to solve problems with menu entries not visible in the screen.
- The form widget geometry code was changed to solve problems with integer round problems in the child widgets geometry when resizing the parent form widget.
- Several bugs were fixed in the text code, while some code was rewritten from scratch.

4.14 Xpm

Version 3.4k of the Xpm (X pixmap) library is now integrated into XFree86.

4.15 xedit

Xedit have been changed to use most of the new features added to the new version of the Xaw library, and some xedit only features were added. Emacs users will find that several of the emacs key bindings work with the new version of xedit. These include:

- File name tab completion. Including a *Emacs dired* like window, that will be shown when there are more than one match, when `C-x,d` is pressed, or when a directory name is specified.
- An unlimited number of files can be edited at the same time. Including multiple views of the same or different files.
- The line number of the cursor position is always visible. It can also be customized to show the column number, the position offset and the current size of the file.
- There is an `autoReplace` resource, that enables automatic text replacement at the time text is typed. This feature is useful to create simple macros, or to correct common spelling errors.
- A fully featured ispell interface is also available. This interface is expected to provide most of the features of the terminal interface of the ispell program, with some extra features that include:
  - A compile time limit of 16 undo levels.
  - Terse mode switch.
  - Dictionary change.
  - The interface also checks for repeated words.
- A first tentative to add programming modes was done. Currently, there is one mode:
  - **C-mode:** this mode is expected to be stable, and fully usable.

4.16 Font support

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

4.17 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.

4.18 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.

4.19 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.

4.20 Large font optimisation

The glyph metrics array, which all the X clients using a particular font have access to, is now 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.

4.21 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.

4.22 Lucidux fonts from Bigelow and Holmes

XFree86 now includes the ''Lucidux'' family of professionally hinted Type 1 fonts. This family consists of the fonts ''Lucidux Serif'', ''Lucidux Sans'' and ''Lucidux Mono'' in Roman and oblique variants, and includes over 370 glyphs in each font covering among others the glyphs needed for ISO 8859-1, 2, 3, 4, 9 and 15. Bold variants will be included in a future release. The design and font outlines were donated by Charles Bigelow and Kris Holmes from Bigelow and Holmes Inc., and the hinting was donated by Berthold Horn and Blenda Horn from Y&Y, Inc. For more information, please contact <design@bigelowandholmes.com> or <sales@yandy.com>, or consult Y&Y's web site `<URL:http://www.yandy.com>`.

4.23 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 1.58 2000/12/17 23:01:10 dawes