Release Notes for XFree86™ 4.0

The XFree86 Project, Inc

7 March 2000

Abstract

This document contains some information about the features present in XFree86 4.0 and their status.

1. Introduction

XFree86 4.0 is the first official release of the new XFree86 4. XFree86 4 represents a significant redesign of the XFree86 X server. It is very important to keep in mind that XFree86 4 is still very much in development, and it contains a lot of new work. That means two things: there is a lot of new exciting stuff to try, but being new code, it hasn't had nearly as much of a workout as the stable 3.3.x releases. If you're looking for a well-tested, stable release, and can't afford the inconveniences that new software can sometimes cause, then you are probably better off sticking with the 3.3.x releases for now. If you have the resources to try out the new version and investigate its features, or if you just like being on the bleeding edge, then please try 4.0!

This release isn't quite as complete as we would have liked. The main missing pieces are a nice configuration tool and support for some of the hardware that 3.3.x supports. The first point means that configuring the server might be more painful than usual. The second means that your hardware might not be supported by 4.0, or it might be supported at a lesser level (conversely, some hardware is better supported in 4.0). We've attempted to provide some information about the second point in our Driver Status document. Please check there first before trying 4.0. Unfortunately that document is still fairly basic, but it should at least give you an idea of whether you're likely to be able to use 4.0 at all or not.

On the subject of configuration, we have updated the basic text-based tool "`xf86config`" to generate config files in the format required by 4.0 (3.3.x config files won't really work with 4.0). We're also working on some other configuration tools, including one that is built-in to the X server. An early version of this is included in the release, and it works well for some hardware. To try it out, just run (as root) "`XFree86 -configure`". Both of these configuration options will at worst give you a reasonable starting point for a suitable configuration file. We've put some effort into documenting the 4.0 config file format, and you can find that information in the XF86Config manual page. Please check that and the driver manual pages and related documentation for further information about that.

Oh, another thing you might notice is that our documentation is rather patchy. Most of what is present should be in reasonable shape, but there are gaps. We thought it better to leave out docs that were very out of date rather than providing inaccurate and misleading information.

Finally, before you download and install the binary distributions for this release, please have a quick read through the Installation Document. It may save you some time.

If those cautionary notes haven't turned you away (and we certainly hope not), please read on... The sections below describe some of the new features and changes between 3.3.x and 4.0. There is a lot of new stuff, and we definitely don't have enough space to cover it all here.

2. X server

Unlike XFree86 3.3.x where there are multiple X server binaries, each of which drive different hardware, XFree86 4.0 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 4.0 has X server support for most UNIX(R) and UNIX-like operating systems on Intel/x86 platforms, plus support for Linux on Alpha and PowerPC platforms. Work on support for additional architectures and operating systems is in progress, and is planned for future releases.

2.1 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, for example, 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 is that when modules are updated, they don't need to be recompiled for each different operating system. We're planning to take advantage of this to provide more frequent driver module updates in between major releases.

The loader in version 4.0 has support for Intel (x86), Alpha and PowerPC platforms. It also has preliminary support for Sparc platforms but this isn't used yet.

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'll 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 can't guarantee that this will be the case. We are planning to fully document and stabilise the module interfaces in a future release, and at that point backward compatibility will be easier to achieve.

Note about module security

> The XFree86 X server runs with root privileges, which means that 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. We hope to have a mechanism for signing/verifying the modules that we provide available in a future release.

2.2 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 deprecated 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"
    Option     "PCI Retry"
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.

2.3 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 ver-
bosity 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.
```

2.4 XAA

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

2.5 Multi-head

Some multi-head configurations are supported in this release, 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).

2.6 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 this release was completely rewritten for improved performance and correctness.

Known problems:

- Most (all?) 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.

2.7 XVideo extension

The XVideo extension is included in this release, but nobody seems interested in writing up some information about it.

2.8 DGA version 2

DGA 2.0 is nearly completed but still not implemented by all drivers. Preliminary documentation for the client libraries can be found in the xc/programs/Xserver/hw/xfree86/DGA document. Some degree of backwards compatibility with version 1.0 is provided. This information is out of date.

2.9 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.

2.10 GLX and the Direct Rendering Infrastructure (DRI)

Precision Insight <URL:http://www.precisioninsight.com> has been 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 and Voodoo3 graphics cards. Additional direct rendering drivers will be available for 3Dfx, Intel, ATI and Matrox boards during the second quarter of 2000. 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>.

2.11 X-Video Extension (Xv)

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. XvImages are only supported by the Matrox G200/G400 cards at the moment.

2.12 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.

2.13 Drivers

XFree86 4.0 includes the following drivers:

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

2.13.1 APM

This is the driver for Alliance AT3D/AT25 and AT24 chips. There is a rather complete support for the functions with acceleration at 8,15,16,24 and 32 bits (limited by the chip at 24bpp). There is preliminary, still buggy, support for the AP6422 chip, which is still supported in 3.3.x servers. The Xv driver is almost ok. The Rush extension for glide2x works, with some additions, including overlay of the result. DGA and DGA2 have been tested ok. Further information can be found in README.apm.

Driver Name	Description
apm	Alliance Pro Motion
ati	ATI
chips	Chips & Technologies
cirrus	Cirrus Logic
cyrix (*)	Cyrix MediaGX
fbdev	Linux fbdev
glide	Glide2x (3Dfx)
glint	3Dlabs, TI
i740	Intel i740
i810	Intel i810
mga	Matrox
neomagic	NeoMagic
nv	NVIDIA
r128	ATI Rage 128
rendition	Rendition
s3virge	S3 ViRGE
sis	SiS
tdfx	3Dfx
tga	DEC TGA
trident	Trident
tseng	Tseng Labs
vga	Generic VGA

2.13.2 Chips & Technologies

Information about the C&T driver can be found in README.chips.

2.13.3 s3virge

The s3virge driver is a port of the 3.3.x SVGA S3 ViRGE driver. As such it should be as stable and functional as previous XFree86 releases. There are a couple additional benefits included primarily due to common enhancements:

- Depth 24 problems resolved with clients using 24/32 bpp pixmaps.
- Our common acceleration architecture (XAA) has been re-written, as has the ViRGE acceleration code. You should find this version has better performance than prior releases.
- Multi-head is reported to work.
- The s3virge man page lists options and has configuration notes for this release of the driver.
- Trio 3D and Trio 3D/2X support has been added, matching the 3.3.6 driver.
- Supports screen rotation and shadow framebuffer.

Outstanding items not implemented or fully tested:

- DGA support is implemented, but preliminary and untested.

Further information can be found in README.s3virge.

2.13.4 TGA

The TGA driver is now accelerated and supports both 8 and 32 plane framebuffers. It is known to work under Linux/Alpha. Please see the README.DECtga file for further information.

2.13.5 Matrox

The MGA driver supports the same range or hardware as XFree86 3.3.4, but has a number of enhancements including multi-head support and support for (non-destructive) overlays (8-bit + 24-bit).

Option `"overlay"` when the server is started in 32bpp (`-fbbpp 32`) will enable the 8+24 mode. The current implementation doesn't optimize away unnecessary exposures yet so the performance of this option will be better in future release. By default, the color key for the overlays is 255, but this can be changed with the `"ColorKey"` option to work around problems in specific programs. Valid values for the key are 2-255.

This release contains performance enhancements for the G400 and particularly for the G400 MAX. It also includes XvImage support for G200/G400 chips and improved memory autodetection support.

Further information can be found in the `mga` man page.

2.13.6 ATI

Information about the ATI driver can be found in README.ati `<URL:ati.html>`. The current version is not accelerated. Acceleration support is planned for a future release.

2.13.7 NVIDIA

The "nv" driver supports all Riva TNT accelerators as well as the new GeForce and Quadro accelerators. DGA 2.0 support is included.

Further information can be found in the nv man page.

2.13.8 Glide

This driver is for Voodoo 1 and Voodoo 2 boards. It runs X on top of the 3DFX Glide API (where this is available, like for Linux). You need to have Glide 2.x installed before you can run this driver. This driver uses no hardware acceleration (since there is no 2D acceleration in these boards) but is rather quick anyway since the CPU renders to local RAM which is then copied block-wise to the board. Unfortunately the Voodoo 1/2 boards are rather limited in resolution. The Voodoo 1 can do 800x600 and the Voodoo 2 can do 1024x768 at best, but still it has some use as a second head in Xinerama or multihead mode.

16 and 24 bpp modes are supported (24 bit in 32-bit sparse-packed mode).

Further information about this driver can be found in the 'glide' driver man page for XFree86. You will not get this driver running before reading this man page.

For Voodoo Banshee and Voodoo 3 boards or later: Please use the tdfx driver which talks directly to the hardware and is much faster.

2.13.9 GLINT

The "glint" driver supports most 3Dlabs/Texas Instruments GLINT/Permedia chips. There is a rather complete support (better than in 3.3.x) for acceleration at 8, 15, 16, and 24 bit depths (limited by some chips at some depths). 8+24 overlay is supported. The Xv extension is supported for some boards.

Further information about this driver can be found in the 'glint' driver man page.

3. X libraries and clients

3.1 Xaw

Two versions of the Xaw library are provided in this release. 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.

3.2 Xpm

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

3.3 xterm

New Features:

- Support Unix98 PTY's.
- Support Unicode using UTF-8 input and output. There are a few limitations, this work is still in progress:
  - You must use the `-u8` command line option to use this feature, as well as compile with the `OPT_WIDE_CHARS` definition. (The feature is compiled when using imake).
  - Input (from keyboard) and output (select/paste) are in UTF-8 form. There is no support in Xlib for UTF-8; xterm uses a lookup table to map keysym codes. Select/paste is done either via `STRING` or using the new atom `UTF8_STRING`.
- Add optional feature (resource and command-line options) to make xterm use the PTY's sense of erase character on startup, rather than requiring it to be `\177`, or set the PTY's erase character to match xterm's configuration. Note that while `$TERMCAP` is modified to reflect the actual configuration, the terminfo `kdch1` string is not. (This feature is also in XFree86 3.3.4).
- Revised keyboard handling, making two modes (VT220 and Sun/PC) which are switched by popup menu. This makes the numeric keypad work as expected. Codes sent by the backarrow and delete keys also are affected.
- Add parameters to function key escape sequences to indicate if shift, control or alt are set. This works for Sun/PC keyboard mode.
- Separated command-line and menu settings for reverse video from that done under program control. This is a problem which was introduced by X11R6. Though correct, most users are confused by allowing the reset command to undo the effect of the command-line `-rv` option.
- Blinking cursor can be specified by resource or popup menu.
- New control sequences for switching between normal and alternate screens maintain separate cursor-save locations for the two screens, avoiding incorrect cursor placement on exit from vi.
- Support line-drawing characters when the font does not include them by drawing them.
- Add support for switching font sizes, by stepping through the font menu using shifted keypad plus and minus.
- New resource `trimSelection` allows xterm to trim trailing blanks from selected lines.
- Provide user applications a means of determining the version of xterm for feature comparison by returning the patch number (e.g., 111) in the secondary DA response.
- Modify treatment of `XK_Delete` keysym so it transmits parameterized VT220-style "`<esc>[3~`" if modifiers (shift, control alt) are given.
- Add ``cacheDoublesize'' resource to limit the caching of font information for double-sized characters, to accommodate X terminals with limited font memory.
- Add ``metaSendsEscape'' resource, with corresponding control sequence and menu entry. Like ``eightBitInput'', this causes xterm to send ESC prefixing the given key, but applies to all keys and is independent of the 8-bit/7-bit terminal setting.
- Implement an 88-color model for systems where 256-colors cannot be allocated.
- Add support for DEC Locator control sequences for xterm. This allows the xterm mouse to be used with applications that use the DEC Locator sequences, such as VAX Tpu, or SMG$ based applications.

- Implement `-hold` option, allowing users to retain the window after a shell has exited.
- Add an application resource, ‘‘messages’’ (and a corresponding `-/+mesg` option) which controls the initial permission on the terminal.
- Implement UTF-8 translation for Media Copy (print) operations.
- Implement vt320 control sequences for Print Composed Main Display and for Print All Pages. The latter directs xterm to print the current screen as well as the scrollback buffer.

Bug fixes/improvements:

- If `colorMode` is enabled by default, compile-in default resources to match the colors listed in `XTerm-col.ad`.
- Deprecate DA answerback string, making it settable via a resource value for applications which may need this.
- Input characters which are mapped when in vt220 National Replacement Character mode.
- Completed support for double size characters.
- Remove `kfnd/kll/kslt` strings from terminfo, because curses applications do not necessarily return `khome/kend` pairs.
- Corrected ifdef’s for menus, which did not allow tek4014 to be suppressed properly.
- Improved tests for determining if xterm should use overstriking to simulate bold fonts.
- Add test/demo scripts for double size characters, font switching, screen resizing and colors.
- Amend treatment of ALT key so that if ALT is used as a modifier in key translations, then no parameter will be sent in escape sequences for Sun/PC function keys.
- Improved the ptyInitialErase logic to make it work better with a terminfo library.
- Modify treatment of line-drawing characters in UTF-8 mode so that Unicode values are used rather than characters 1-31 for storing the translated characters.
- Modify translation of UTF-8 sequences to reject ‘‘overly long’’ variations.
- Correct a case where colors were not rendered properly. This happened when an application inserted several lines, then changed colors. If this was done all in one write, then there would be no intervening refresh, and the new color was applied to the pending scrolling operation which was awaiting the next refresh.
- Corrected misspelled resource name in command-line option for HP function keys.
- Change label on ‘‘Sun/PC Keyboard’’ popup menu entry to ‘‘VT220 Keyboard’’, since the checked state corresponds to VT220 rather than Sun/PC.
- Two corrections to simulation of bold font via overstriking:
  - use clipping to avoid leaving trash at end of the text, and
  - add brackets so wide-character logic does not fall-through into the overstriking logic.
- Modify checks for repeat-character control sequence to test the character class against xterm’s state table, rather than the `isprint()` macro.
- Modify terminfo entry for ‘‘xterm-xfree86’’ to reflect modifiers for shift and control.
- Add several entries to termcap file to make it have the same set of aliases as the terminfo file.

- Scale the color values used for xterm-256color terminfo entry to 0..1000, as expected by ncurses.
- Change xterm-r6 terminfo definitions for F1-F4 to match program.
- Remove obsolete documentation about modifiers which can be returned in mouse tracking mode, and modify logic to ignore modifiers other than the existing ones, e.g., NumLock.
- Use free bit from obsolete shift-modifier coding of mouse tracking button events to encode buttons 4 and 5, e.g., for a wheel mouse. Move the suggested wheel-mouse button translations into charproc.c to simplify customization.
- Modify warning if change-ownership of PTY fails; some configurations may not happen to have old-style PTY's.
- Add more information, i.e., with strerror for some system calls in the main program which may fail due to insufficient permissions.
- Various improvements to configure script, e.g., tests for utmp.

3.4 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. Fonts and Internationalisation

Details about the font support in this version of XFree86 can be found in the README.fonts document.

4.1 TrueType support

This version of XFree86 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.

4.2 CID font support

Support for CID-keyed fonts is included in this version of XFree86. 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.

4.3 Internationalisation of the scalable font backends

A new "fontenc" layer has been added 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.

4.4 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.

4.5 Unicode/ISO 10646 support

What is included:

- 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.

Known problems:

- Xlib does not yet fully support UTF-8 as a locale, which means that xterm UTF-8 keyboard support is at the moment a temporary hack.
- Most ISO10646-1 fonts encode no characters above U+31FF. This avoids the inefficient allocation and transmission of a >700 kB large XFontStruct structure, which would happen if the (not very important) ligatures and symbols above U+f000 were present.
- ISO 10646 Level 2 combining characters are not yet supported by xterm (will be needed for instance for Thai and IPA).
- Switching between a half-width and full-width font pair (such as 9x18 and 18x18ja) is not yet supported by xterm (will be needed for CJK scripts).

4.6 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>`.

5. Miscellaneous

5.1 Directory rearrangements

Some changes to the installed XFree86 directory structure have been implemented for 4.0. 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 1.37 2000/03/08 20:11:55 dawes`