Release Notes for X11R7.5

The X.Org Foundation

October 2009

These release notes contain information about features and their status in the X.Org Foundation X11R7.5 release.

Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to the X11R7.5 Release</td>
<td>3</td>
</tr>
<tr>
<td>Summary of new features in X11R7.5</td>
<td>3</td>
</tr>
<tr>
<td>Overview of X11R7.5</td>
<td>4</td>
</tr>
<tr>
<td>Details of X11R7.5 components</td>
<td>5</td>
</tr>
<tr>
<td>Build changes and issues</td>
<td>10</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>11</td>
</tr>
<tr>
<td>Deprecated components and removal plans</td>
<td>12</td>
</tr>
<tr>
<td>Attributions/Acknowledgements/Credits</td>
<td>13</td>
</tr>
</tbody>
</table>
Introduction to the X11R7.5 Release

This release is the sixth modular release of the X Window System. The next full release will be X11R7.6 and is expected in 2010.

Unlike X11R1 through X11R6.9, X11R7.x releases are not built from one monolithic source tree, but many individual modules. These modules are distributed as individual source code releases, and each one is released when it is ready, instead of only when the overall window system is ready for release. The X11R7.x releases are made by “rolling up” the individual module releases into a collection that is often affectionately called the “katamari” by the developers.

The X11R7.5 release does not include all of the software formerly included in the previous X Window System releases. It is designed to be a reasonable baseline from which to start when building the window system for the first time for a new installation, distribution, or package set. It does not provide a full desktop environment, expecting a more feature rich set of applications to be installed from one of the several excellent desktop environments available for the X Window System. The X.Org developers continue to maintain and produce new releases of much of the software that was formerly in the main window system releases but is no longer included in the katamari releases, including many of the Athena Widgets desktop applications that were provided as samples in previous window system versions.

Once their window system build is established, most builders watch for announcements of individual module updates on the xorg-announce mailing list and update to those as needed. The X.Org Foundation currently releases the X Window System katamari releases approximately once a year, but many modules, especially the X servers and drivers, are updated more frequently between those releases.

For help with how to build and develop in the modular tree see the Modular Developer’s Guide in the X.Org wiki.

We encourage you to submit bug fixes and enhancements to freedesktop.org’s bug tracking system using the xorg product, and to discuss them on xorg@lists.freedesktop.org. More details on patch submission and review process are available on the SubmittingPatches page of the X.Org wiki.

The release numbering is based on the original MIT X numbering system. X11 refers to the version of the network protocol that the X Window system is based on: Version 11 was first released in 1988 and has been stable for 21 years, with only upward compatible additions to the core X protocol, a record of stability envied in computing. Formal releases of X started with X version 9 from MIT; the first commercial X products were based on X version 10. The MIT X Consortium and its successors, the X Consortium, the Open Group X Project Team, and the X.Org Group released versions X11R3 through X11R6.6. Since the founding of the X.Org Foundation in early 2004, many further releases have been issued, from X11R6.7 to the current 7.5.

The next section describes what is new in the latest version (7.5) compared with the previous full release (7.4).

Summary of new features in X11R7.5

This is a sampling of the new features in X11R7.5. A more complete list of changes can be found in the ChangeLog files that are part of the source of each X module.

- **Multi-Pointer X (MPX)** provides the user with multiple independent mouse cursors and multiple independent keyboard foci. Each cursor is a true system cursor and different pointers can operate in multiple applications simultaneously.

- **Input device properties** allow you to attach properties to a device. These properties can be of arbitrary type and can be changed without the server having to know their details.
The X Input Extension version 2.0 (XI2) is designed to replace both core input processing and prior versions of the X Input Extension. Besides MPX, it provides a number of other enhancements over version 1.5, including:
- use of XGE and GenericEvents.
- explicit device hierarchy of master and slave devices.
- the ability for devices to change capabilities at runtime.
- raw device events

Resize, Rotate and Reflect Extension (RANDR) version 1.3 builds on the changes made with version 1.2 and adds some new capabilities without fundamentally changing the extension again. The following features are added in this version:

Projective Transforms
The implementation work for general rotation support made it trivial to add full projective transformations. These can be used to scale the screen up/down as well as perform projector keystone correct or other effects.

Panning
Panning was removed with RandR 1.2 because the old semantics didn’t fit any longer. With RandR 1.3 panning can be specified per crtc.

The DRI2 extension is designed to associate and access auxiliary rendering buffers with an X drawable. It is a essentially a helper extension to support implementation of direct rendering drivers/libraries/technologies. The first consumer of this extension is a direct rendering OpenGL driver, but the DRI2 extension is not designed to be OpenGL specific. Work is underway to utilize DRI2 for the Video Decode and Presentation API for Unix (VPDAU) as well. Direct rendering implementations of OpenVG, Xv, cairo and other graphics APIs should find the functionality exposed by this extension helpful and hopefully sufficient.

Video and input driver enhancements. Please see the ChangeLog files for individual drivers; there are far too many updates to list here.

... and the usual assortment of correctness and crash fixes.

Overview of X11R7.5
On most platforms, X11R7.5 has a single hardware-driving X server binary called Xorg. This binary can dynamically load the video drivers, input drivers, and other modules that are needed. Xorg has currently has support for Linux, Solaris, and some BSD OSs on Alpha, PowerPC, IA-64, AMD64, Intel x86, Sparc, and MIPS platforms.
Additional specialized X server binaries may be found depending on the platform and build configuration, including:

Xdmx
is a proxy X server that uses one or more other X servers as its display devices. It provides multi-head X functionality for displays that might be located on different machines.

Xnest
is a nested X server, that operates as both an X client and X server. Xnest is a client of the real server which manages windows and graphics requests on its
behalf. Xnest is a server to its own clients, and manages windows and graphics requests on their behalf. To these clients, it appears to be a conventional server.

**Xephyr**

is a X server that outputs to a window on a pre-existing “host” X display. Unlike Xnest which is an X proxy, and thus limited to the capabilities of the host X server, Xephyr is a full X server which uses the host X server window as a “framebuffer” via fast SHM XImages.

**Xvfb**

is a virtual framebuffer X server that can run on machines with no display hardware and no physical input devices. It emulates a dumb framebuffer using virtual memory.

**Xquartz**

is an X server that interacts with the MacOS X native Aqua window system, displaying windows on the Mac desktop and accepting input from the Mac system devices, allowing X11 applications to be used in a native Mac desktop session.

**Xwin**

is an X server that runs under the Cygwin environment, interacting with the Microsoft Windows native window system, displaying windows on the Windows desktop and accepting input from the Windows system devices, allowing X11 applications to be used in a native Windows desktop session.

### Details of X11R7.5 components

#### Video Drivers

X11R7.5 includes the following video drivers:

<table>
<thead>
<tr>
<th>Driver Name</th>
<th>Description</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>apm</td>
<td>Alliance Pro Motion</td>
<td>README.apm</td>
</tr>
<tr>
<td>ark</td>
<td>Ark Logic</td>
<td></td>
</tr>
<tr>
<td>ast</td>
<td>ASPEED Technology</td>
<td></td>
</tr>
<tr>
<td>chips</td>
<td>Chips &amp; Technologies</td>
<td>README.chips, chips(4)</td>
</tr>
<tr>
<td>cirrus</td>
<td>Cirrus Logic</td>
<td></td>
</tr>
<tr>
<td>fbdev</td>
<td>Linux framebuffer device</td>
<td>fbdev(4)</td>
</tr>
<tr>
<td>geode (*)</td>
<td>AMD Geode GX and LX</td>
<td></td>
</tr>
<tr>
<td>Package</td>
<td>Vendor/Description</td>
<td>Notes</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>glint</td>
<td>3Dlabs, TI</td>
<td>glint(4)</td>
</tr>
<tr>
<td>i128</td>
<td>Number Nine</td>
<td>README.i128, i128(4)</td>
</tr>
<tr>
<td>i740</td>
<td>Intel i740</td>
<td>README.i740</td>
</tr>
<tr>
<td>imstt</td>
<td>Integrated Micro Solns</td>
<td></td>
</tr>
<tr>
<td>intel</td>
<td>Intel i8xx/i9xx</td>
<td>README.intel, intel(4)</td>
</tr>
<tr>
<td>mach64</td>
<td>ATI Mach64</td>
<td>README.ati</td>
</tr>
<tr>
<td>mga</td>
<td>Matrox</td>
<td>mga(4)</td>
</tr>
<tr>
<td>neomagic</td>
<td>NeoMagic</td>
<td>neomagic(4)</td>
</tr>
<tr>
<td>newport</td>
<td>SGI Newport</td>
<td>README.newport, newport(4)</td>
</tr>
<tr>
<td>nsc</td>
<td>National Semiconductor</td>
<td>nsc(4)</td>
</tr>
<tr>
<td>nv</td>
<td>NVIDIA</td>
<td>nv(4)</td>
</tr>
<tr>
<td>r128</td>
<td>ATI Rage128</td>
<td>README.r128, r128(4)</td>
</tr>
<tr>
<td>radeon</td>
<td>ATI Radeon</td>
<td>radeon(4)</td>
</tr>
<tr>
<td>rendition</td>
<td>Rendition</td>
<td>README.rendition, rendition(4)</td>
</tr>
<tr>
<td>s3</td>
<td>S3 (not ViRGE or Savage)</td>
<td></td>
</tr>
<tr>
<td>s3virge</td>
<td>S3 ViRGE</td>
<td>README.s3virge, s3virge(4)</td>
</tr>
<tr>
<td>savage</td>
<td>S3 Savage</td>
<td>savage(4)</td>
</tr>
<tr>
<td>siliconmotion</td>
<td>Silicon Motion</td>
<td>siliconmotion(4)</td>
</tr>
<tr>
<td>sis</td>
<td>SiS</td>
<td>README.SiSiS, sis(4)</td>
</tr>
<tr>
<td>sisusb</td>
<td>SiS USB</td>
<td>sisusb(4)</td>
</tr>
<tr>
<td>suncg14</td>
<td>Sun cg14</td>
<td></td>
</tr>
<tr>
<td>suncg3</td>
<td>Sun cg3</td>
<td></td>
</tr>
<tr>
<td>suncg6</td>
<td>Sun GX and Turbo GX</td>
<td></td>
</tr>
<tr>
<td>sunffb</td>
<td>Sun Creator/3D, Elite 3D</td>
<td></td>
</tr>
<tr>
<td>sunleo</td>
<td>Sun Leo (ZX)</td>
<td></td>
</tr>
<tr>
<td>suntcx</td>
<td>Sun TCX</td>
<td></td>
</tr>
<tr>
<td>tdfx</td>
<td>3Dfx Voodoo Banshee, 3, 4 &amp; 5</td>
<td>tdfx(4)</td>
</tr>
<tr>
<td>tga</td>
<td>DEC TGA</td>
<td>README.DECTga</td>
</tr>
<tr>
<td>trident</td>
<td>Trident</td>
<td>trident(4)</td>
</tr>
<tr>
<td>tseng</td>
<td>Tseng Labs</td>
<td></td>
</tr>
<tr>
<td>v4l</td>
<td>Video4Linux</td>
<td>v4l(4)</td>
</tr>
<tr>
<td>vesa</td>
<td>VESA</td>
<td>vesa(4)</td>
</tr>
<tr>
<td>vmware</td>
<td>VMware guest OS</td>
<td>vmware(4)</td>
</tr>
<tr>
<td>voodoo</td>
<td>3Dfx Voodoo 1 &amp; 2</td>
<td>voodoo(4)</td>
</tr>
<tr>
<td>wsfb</td>
<td>Workstation Framebuffer</td>
<td>wsfb(4)</td>
</tr>
<tr>
<td>xgi</td>
<td>XGI</td>
<td>xgi(4)</td>
</tr>
<tr>
<td>xgixp</td>
<td>XGI XP</td>
<td>xgixp(4)</td>
</tr>
</tbody>
</table>
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.

**Input Drivers**

X11R7.5 includes the following input drivers:

<table>
<thead>
<tr>
<th>Driver Name</th>
<th>Description</th>
<th>Further Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>acecad</td>
<td>Acecad Flair</td>
<td>acecad(4)$^6$</td>
</tr>
<tr>
<td>aiptek(*)</td>
<td>Aiptek USB tablet</td>
<td>aiptek(4)$^6$</td>
</tr>
<tr>
<td>evdev(*)</td>
<td>Linux kernel EvDev</td>
<td>evdev(4)$^6$</td>
</tr>
<tr>
<td>joystick</td>
<td>Joystick</td>
<td>joystick(4)$^6$</td>
</tr>
<tr>
<td>kbd</td>
<td>generic keyboards</td>
<td>kbd(4)$^6$</td>
</tr>
<tr>
<td></td>
<td>(non-evdev systems)</td>
<td></td>
</tr>
<tr>
<td>mouse</td>
<td>most mouse devices</td>
<td>mouse(4)$^6$</td>
</tr>
<tr>
<td></td>
<td>(non-evdev systems)</td>
<td></td>
</tr>
<tr>
<td>synaptics</td>
<td>Synaptics &amp; ALP touchpads</td>
<td>synaptics(4)$^6$</td>
</tr>
<tr>
<td>vmmouse</td>
<td>VMWare virtual mouse</td>
<td>vmmouse(4)$^6$</td>
</tr>
<tr>
<td>void</td>
<td>dummy device</td>
<td>void(4)$^6$</td>
</tr>
</tbody>
</table>

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

**Xorg server**

**Loader and Modules**

The Xorg server relies on the operating system’s native module loader support for handling program modules. The X server makes use of modules for video drivers, X server extensions, input device drivers, framebuffer layers, and internal components used by some drivers (like XAA & EXA).

The module interfaces (both API and ABI) used in this release are subject to change without notice. While we will attempt to provide backward compatibility for the module interfaces, we cannot guarantee this. Compatibility in the other direction is explicitly not guaranteed because new modules may rely on interfaces added in new releases.

**Note about module security**

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

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

Note that this release features significant improvements for running the server without a configuration file, so many users may find that they don’t need a configuration file.

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

The recommended method for generating a configuration file is to use the Xorg server itself. Run as root:

```
Xorg -configure
```

and follow the instructions.

Command Line Options

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

XAA

The XFree86 Acceleration Architecture (XAA) was completely rewritten from scratch for XFree86 4.x and is used in X11R7.5. Most drivers implement acceleration by making use of the XAA module.

EXA

EXA was created as a new driver acceleration architecture to replace XAA. EXA was designed specifically to accelerate Render operations. This release features improved driver support for EXA. See the individual driver changelogs for details.

Multi-head

Some multi-head configurations are supported in X11R7.5. Support for multiple PCI/AGP cards may require a kernel with changes to support VGA arbitration.

One of the main problems is with drivers not sufficiently initializing cards that were not initialized 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).

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 \texttt{+xinerama} command line option for the X server.

**DDC**

The VESA® Display Data Channel (DDC™) standard allows the monitor to tell the video card (or in 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: \texttt{Option "NoDDC"}. We have support for DDC versions 1 and 2; these can be disabled independently with \texttt{Option "NoDDC1"} and \texttt{Option "NoDDC2"}.

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

\textit{Changed behavior caused by DDC.}

Several drivers use DDC information to set the screen size and pitch. This can be overridden by explicitly resetting it to the and non-DDC default value 75 with the \texttt{-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.

**GLX and the Direct Rendering Infrastructure (DRI)**

Direct rendered OpenGL® support is provided for several hardware platforms by the Direct Rendering Infrastructure (DRI). Further information about DRI can be found at the DRI Project’s web site\textsuperscript{9}. The 3D core rendering component is provided by Mesa\textsuperscript{10}.

Of note is that this release supports building the X server using the system-wide libdrm. Previously, drm was kept in the server’s tree and loaded as a module, rather than using the standard OS mechanisms for managing shared libraries of code. This requires that the server be built using a version of libdrm of 2.3.0 or newer if it is to use DRM.

**Terminate Server keystroke**

The Xorg server has previously allowed users to exit the server by pressing the keys \texttt{Control + Alt + Backspace}. While this function is still enabled by default in this release, the keymap data usually used with Xorg, from the xkeyboard-config project, has been modified to not map that sequence by default, in order to reduce the chance that inexperienced users will accidentally destroy their work.

Users who wish to have this functionality available by default may enable it via the XKB configuration option \texttt{"terminate:ctrl_alt_bksp"}. For instance, the \texttt{setxkbmap} command can be used to enable this by running:

\begin{verbatim}
setxkbmap -option "terminate:ctrl_alt_bksp"
\end{verbatim}

Many desktop environments include XKB configuration options in their preferences to enable this as well.

\begin{footnotesize}
\begin{itemize}
\item[9] \textsuperscript{9} Web site for the DRI Project: \url{http://www dri.org}
\item[10] \textsuperscript{10} Mesa is a graphics library and rendering framework based on the OpenGL API: \url{http://mesa3d.org}
\end{itemize}
\end{footnotesize}
X Server startup state

The X servers in the X11R7.5 release now start by default with an empty black screen and do not draw the mouse cursor until a client sets the cursor image. To restore the classic behavior of starting with the grey weave pattern and × cursor, start the X server with the --retro option.

Font support

Details about the font support in X11R7.5 can be found in the README.fonts document.

Default font installation directory

Previous versions of X installed font files under the lib/X11/fonts subdirectory of the X installation directory (for instance, in X11R6 releases, /usr/X11R6/lib/X11/fonts was commonly used). This release changes the default installation path to the fonts subdirectory of the datadir setting from the GNU autoconf configuration. For instance, if the fonts are configured with ./configure --prefix=/usr, they will be installed under subdirectories of /usr/share/fonts/X11. The font module configure scripts all take an option of --with-fontrootdir=PATH to override the default. If --with-fontrootdir is not specified, the fontutil pkg-config file will be consulted to find the fontrootdir specified when the fontutil module was installed.

Bitmap font compression methods

The X11R7.5 release supports PCF format bitmap fonts stored uncompressed or compressed via the compress, gzip, or bzip2 programs. To utilize bzip2 compression, the libXfont and mkfontscale modules must be built with the --with-bzip2 — all other methods are enabled by default.

To specify which compression method to use when installing a font module from X11R7.5 the configure scripts accept an option of --with-compression=TYPE, where TYPE may be none, compress, gzip, or bzip2.

Type1 Font support

Previous versions of X came with two Postscript Type1 font backends. The functionality from the “Type1” backend has been replaced by the Type1 support in the “FreeType” backend.

CID Font support

The CID-keyed font format was designed by Adobe Systems for fonts with large character sets. The CID-keyed format is obsolete, as it has been superseded by other formats such as OpenType/CFF and support for CID-keyed fonts has been removed from X11.
Build changes and issues

Silent build rules
Most of the modules in this release use the AM_SILENT_RULES option of GNU automake 1.11. When building the software, most output will show an abbreviated format for the commands being run, such as:

    CC xmen.o

To enable verbose output, showing all the arguments to the commands being run, add the flag \texttt{V=1} to the \texttt{make} command line or add the flag \texttt{--disable-silent-rules} to the \texttt{configure} command.

New configure options for font modules
Several new options have been added to the \texttt{configure} scripts for font modules in this release. See the Font support section of this document for details of the \texttt{--with-fontrootdir=PATH} and \texttt{--with-compression=TYPE} options.

Changes to extension headers
The C language header files for a number of X11 protocol extensions were refactored in this release to better split the protocol definitions and the client library definitions. Efforts were made to retain compatibility for existing software, but use of some headers may now trigger warnings suggesting including new or more appropriate headers instead.

Since these changes were made to files in both the \texttt{proto} and \texttt{lib} modules for each extension, builders upgrading individual modules will have to update these modules in unison to avoid breaking builds of software using the headers from these modules.

Miscellaneous
This section describes other items of note for the X11R7.5 release.

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

    /\texttt{tmp}/.font-unix
    /\texttt{tmp}/.ICE-unix
    /\texttt{tmp}/.X11-unix

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

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

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

**Composite exposes extra visuals**

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

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

**Deprecated components and removal plans**

This section lists current plans for removal of obsolete or deprecated components in the X.Org releases. As our releases are open source, users who continue to require these can find the source in previous releases and continue to use these, but the X.Org Foundation and its volunteers have decided the burden of continued maintenance and distribution in the core X11 releases outweighs the benefits of doing so. In some cases, this is simply because no one has volunteered to do continued maintenance, so if software is listed here that you need, you can contact <xorg@lists.freedesktop.org> to volunteer to take over maintainership, either inside or outside of the Xorg release process.

**Future Removals**

**DGA version 2**

DGA 2.0 is included in 7.5. Documentation for the client libraries can be found in the XDGA(3) man page. DGA should be considered deprecated; if you are relying on it, please let us know what you need it for so we can find better solutions.

**Input device discovery via HAL**

The Xorg server currently uses the HAL framework to discover connected input devices, receive notification of hotplug events for them, and to retrieve configuration parameters for them. The HAL maintainers have deprecated HAL, so the X.Org developers are investigating alternatives. As a result, configuration of input devices via HAL *.fdi files may not be supported in future Xorg server releases.

**Xsdl server**

The experimental Xsdl server has never been finished or maintained, and will be removed in future X server releases.
Removed in this Release

Xprint

The Xprint server and extension have been removed in this release. The libXaw8 variant of the Athena Widgets which added Xprint widgets has been removed from this release. Xprint support in a number of client programs has also been removed.

kdrive servers

The kdrive X servers for vesa, ati, chips, epson, i810, igs, ipaq, itsy, mach64, mga, neomagic, nvidia, pcmcia, pm2, r128, savage, sis300, sis530, smi, trident, trio, ts300, via, and vxworks have been removed in this release. Most of these have not worked or been maintained in recent releases.

Unmaintained extensions

Support has been removed from the X servers for the following extensions, which were obsolete, not widely used, or not working:
• AppGroup
• EVI
• FontCache
• MIT-SUNDRY-NONSTANDARD
• TOG-CUP
• XTrap
• XFree86-Misc
• XEvIE

Xorg configuration utilities

The xorgcfg GUI and xorgconfig CLI utilities have been removed in this release. See the Configuration File section for alternative methods of Xorg configuration.

ioprt

The ioprt utility and its aliases (inb, inw, inl, outb, outw, and outl) for manipulating I/O space addresses directly have been removed in this release.

Attributions/Acknowledgements/Credits

This section lists the credits for the X11R7.5 release. For a more detailed breakdown, refer to the ChangeLog file in the source tree for each module, the history in the xorg product in freedesktop.org’s git repositories or the ‘git log’ information for individual source files.

The X Window System has been a collaborative effort from its inception. Our apologies for anyone or organization inadvertently overlooked. Many individuals (including major contributors) who worked on X are represented by their employers in this list. If you feel we have left anyone out, please let us know.

These people contributed in some way to X11R7.5:

Aaron Plattner  Jordan Crouse
Aaron Zang  Joseph Adams
Release Notes for X11R7.5

George Sapountzis  Søren Hauberg
George Staplin  Søren Sandmann Pedersen
Giuseppe Bilotta  Stefan Dirsch
Gonéri Le Bouder  Stijn van Drongelen
Guillem Jover  Stuart Bennett
Hans de Goede  Stuart Kreitman
Hasso Tepper  Tero Saarni
Havoc Pennington  Theppitak Karoonboonyanan
Helge Bahmann  Thomas Bodzar
Henrik Rydberg  Thomas Jaeger
Henry unbongo  Thomas Klausner
Hong Liu  Thomas Petazzoni
Hugo Jacques  Thorvald Natvig
Ian Romanick  Tiago Vignatti
Imranullah Syed  Tibi Nagy
Ivaylo Boyadzhiev  Tilman Sauerbeck
Jakob Bornecrantz  Timo Aaltonen
Jakub Bogusz  Tom Jaeger
James Cloos  Tomas Carnecky
Jamey Sharp  Tomas Janousek
Jamie Lentin  Topi Kanerva
Jason Vas Dias  Tormod Volden
Jasper Lievisse Adriaanse  vehemens
Jay Cotton  Vincent Mussard
Jeff Smith  walter
Jens Granseuer  Werner LEMBERG
Jens Herden  Will Thompson
Jeremy C. Reed  William Grant
Jeremy Huddleston  Winfried Grünewald
Jeremy Jay  Wolke Liu
Jeremy Lainé  Wu Fengguang
Jeremy Uejio  Xake
Jerome Glisse  Xavier Bestel
Jerome Pinot  Xiang, Haihao
Jesse Adkins  Xue Wei
Jesse Barnes  Y.C. Chen
Jesse Ruffin  Yaakov Selkowitz
Jie Luo  Yan Li
Jim Huang  Yang Zhao
Jochen Voss  Yann Droneaud
Joe Krahn  Yannick Heneault
Joel Bosveld  Yinan Shen
John Hein  <50724><50976><50672>(Yu-yeon Oh)
John McKernan  Zdenek Kabelac
John Nielsen  Zhao Yakui
John Tapsell  Zhenyu Wang
Jon TURNLEY  Zou Nan hai

This product includes software developed by:

2d3d Inc.  Lars Knoll
3Dlabs Inc. Ltd.  Lawrence Berkeley Laboratory
Aaron Plattner  Leif Delgass
Adam de Boor  Lennart Augustsson
Adam Jackson  Leon Shiman
Adobe Systems Inc.  Lexmark International Inc.
Release Notes for X11R7.5

Kazushi (Jam) Marukawa  XFree86 Project Inc.
Kazuyuki (ikko-) Okamoto  Xi Graphics Inc.
Kean Johnston  X-Oz Technologies
Keith Packard  X-TrueType Server Project
Keith Whitwell  Yu Shao
Kensuke Matsuzaki  Zack Rusin
Kevin E. Martin  Zephaniah E. Hull
Kristian Høgsberg  Zhenyu Wang
Larry Wall

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

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

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

This product includes software developed by Christopher G. Demetriou.

This product includes software developed by the NetBSD Foundation, Inc. (http://www.netbsd.org/) and its contributors.

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

Notes

6. xorg.conf.5.html
7. xorg.conf.5.html
8. Xorg.1.html
11. fonts/fonts.html
12. XDGA.3.man