

**NAME**

Xserver – X Window System display server

**SYNOPSIS**

**X** [option ...]

**DESCRIPTION**

X is the generic name for the X Window System display server. It is frequently a link or a copy of the appropriate server binary for driving the most frequently used server on a given machine.

**STARTING THE SERVER**

The X server is usually started from the X Display Manager program [xdm\(1\)](#) or a similar display manager program. This utility is run from the system boot files and takes care of keeping the server running, prompting for usernames and passwords, and starting up the user sessions.

Installations that run more than one window system may need to use the [xinit\(1\)](#) utility instead of a display manager. However, xinit is to be considered a tool for building startup scripts and is not intended for use by end users. Site administrators are **strongly** urged to use a display manager, or build other interfaces for novice users.

The X server may also be started directly by the user, though this method is usually reserved for testing and is not recommended for normal operation. On some platforms, the user must have special permission to start the X server, often because access to certain devices (e.g. */dev/mouse*) is restricted.

When the X server starts up, it typically takes over the display. If you are running on a workstation whose console is the display, you may not be able to log into the console while the server is running.

**OPTIONS**

Many X servers have device-specific command line options. See the manual pages for the individual servers for more details; a list of server-specific manual pages is provided in the SEE ALSO section below.

All of the X servers accept the command line options described below. Some X servers may have alternative ways of providing the parameters described here, but the values provided via the command line options should override values specified via other mechanisms.

***:displaynumber***

The X server runs as the given *displaynumber*, which by default is 0. If multiple X servers are to run simultaneously on a host, each must have a unique display number. See the DISPLAY NAMES section of the X(7) manual page to learn how to specify which display number clients should try to use.

***–a number***

sets pointer acceleration (i.e. the ratio of how much is reported to how much the user actually moved the pointer).

***–ac***

disables host-based access control mechanisms. Enables access by any host, and permits any host to modify the access control list. Use with extreme caution. This option exists primarily for running test suites remotely.

***–audit level***

sets the audit trail level. The default level is 1, meaning only connection rejections are reported. Level 2 additionally reports all successful connections and disconnects. Level 4 enables messages from the SECURITY extension, if present, including generation and revocation of authorizations and violations of the security policy. Level 0 turns off the audit trail. Audit lines are sent as standard error output.

***–auth authorization-file***

specifies a file which contains a collection of authorization records used to authenticate access. See also the [xdm\(1\)](#) and [Xsecurity\(7\)](#) manual pages.

***–background none***

Asks the driver not to clear the background on startup, if the driver supports that. May be useful for smooth transition with eg. fbdev driver. For security reasons this is not the default as the

- screen contents might show a previous user session.
- br** sets the default root window to solid black instead of the standard root weave pattern. This is the default unless **-retro** or **-wr** is specified.
  - bs** disables backing store support on all screens.
  - c** turns off key-click.
  - c volume** sets key-click volume (allowable range: 0-100).
  - cc class** sets the visual class for the root window of color screens. The class numbers are as specified in the X protocol. Not obeyed by all servers.
  - core** causes the server to generate a core dump on fatal errors.
  - displayfd fd** specifies a file descriptor in the launching process. Rather than specify a display number, the X server will attempt to listen on successively higher display numbers, and upon finding a free one, will write the display number back on this file descriptor as a newline-terminated string. The **-pn** option is ignored when using **-displayfd**.
  - deferglyphs whichfonts** specifies the types of fonts for which the server should attempt to use deferred glyph loading. *whichfonts* can be all (all fonts), none (no fonts), or 16 (16 bit fonts only).
  - dpi resolution** sets the resolution for all screens, in dots per inch. To be used when the server cannot determine the screen size(s) from the hardware.
  - dpms** enables DPMS (display power management services), where supported. The default state is platform and configuration specific.
  - dpms** disables DPMS (display power management services). The default state is platform and configuration specific.
  - extension extensionName** disables named extension. If an unknown extension name is specified, a list of accepted extension names is printed.
  - +extension extensionName** enables named extension. If an unknown extension name is specified, a list of accepted extension names is printed.
  - f volume** sets beep (bell) volume (allowable range: 0-100).
  - fc cursorFont** sets default cursor font.
  - fn font** sets the default font.
  - fp fontPath** sets the search path for fonts. This path is a comma separated list of directories which the X server searches for font databases. See the FONTS section of this manual page for more information and the default list.
  - help** prints a usage message.
  - I** causes all remaining command line arguments to be ignored.
  - maxbigreqsize size** sets the maximum big request to *size* MB.

- nocursor**  
disable the display of the pointer cursor.
- nolisten** *trans-type*  
disables a transport type. For example, TCP/IP connections can be disabled with **–nolisten tcp**. This option may be issued multiple times to disable listening to different transport types.
- noreset**  
prevents a server reset when the last client connection is closed. This overrides a previous **–terminate** command line option.
- p** *minutes*  
sets screen-saver pattern cycle time in minutes.
- pn**  
permits the server to continue running if it fails to establish all of its well-known sockets (connection points for clients), but establishes at least one. This option is set by default.
- nopn**  
causes the server to exit if it fails to establish all of its well-known sockets (connection points for clients).
- r**  
turns off auto-repeat.
- r**  
turns on auto-repeat.
- retro**  
starts the stipple with the classic stipple and cursor visible. The default is to start with a black root window, and to suppress display of the cursor until the first time an application calls `XDefineCursor()`. For the Xorg server, this also sets the default for the `DontZap` option to `FALSE`. For kdrive servers, this implies `-zap`.
- s** *minutes*  
sets screen-saver timeout time in minutes.
- su**  
disables save under support on all screens.
- seat** *seat*  
seat to run on. Takes a string identifying a seat in a platform specific syntax. On platforms which support this feature this may be used to limit the server to expose only a specific subset of devices connected to the system.
- t** *number*  
sets pointer acceleration threshold in pixels (i.e. after how many pixels pointer acceleration should take effect).
- terminate**  
causes the server to terminate at server reset, instead of continuing to run. This overrides a previous **–noreset** command line option.
- to** *seconds*  
sets default connection timeout in seconds.
- tst**  
disables all testing extensions (e.g., `XTEST`, `XTrap`, `XTestExtension1`, `RECORD`).
- ttyxx**  
ignored, for servers started the ancient way (from `init`).
- v**  
sets video-off screen-saver preference.
- v**  
sets video-on screen-saver preference.
- wm**  
forces the default backing-store of all windows to be `WhenMapped`. This is a backdoor way of getting backing-store to apply to all windows. Although all mapped windows will have backing store, the backing store attribute value reported by the server for a window will be the last value established by a client. If it has never been set by a client, the server will report the default value, `NotUseful`. This behavior is required by the X protocol, which allows the server to exceed the client's backing store expectations but does not provide a way to tell the client that it is doing so.

- wr** sets the default root window to solid white instead of the standard root weave pattern.
- x *extension***  
loads the specified extension at init. This is a no-op for most implementations.
- [+/-]xinerama**  
enables(+) or disables(-) the XINERAMA extension. The default state is platform and configuration specific.

## SERVER DEPENDENT OPTIONS

Some X servers accept the following options:

- ld *kilobytes***  
sets the data space limit of the server to the specified number of kilobytes. A value of zero makes the data size as large as possible. The default value of -1 leaves the data space limit unchanged.
- lf *files*** sets the number-of-open-files limit of the server to the specified number. A value of zero makes the limit as large as possible. The default value of -1 leaves the limit unchanged.
- ls *kilobytes***  
sets the stack space limit of the server to the specified number of kilobytes. A value of zero makes the stack size as large as possible. The default value of -1 leaves the stack space limit unchanged.
- render *default|mono|gray|color*** sets the color allocation policy that will be used by the render extension.
  - default* selects the default policy defined for the display depth of the X server.
  - mono* don't use any color cell.
  - gray* use a gray map of 13 color cells for the X render extension.
  - color* use a color cube of at most 4\*4\*4 colors (that is 64 color cells).
- dumbSched**  
disables smart scheduling on platforms that support the smart scheduler.
- schedInterval *interval***  
sets the smart scheduler's scheduling interval to *interval* milliseconds.

## XDMCP OPTIONS

X servers that support XDMCP have the following options. See the *X Display Manager Control Protocol* specification for more information.

- query *hostname***  
enables XDMCP and sends Query packets to the specified *hostname*.
- broadcast**  
enable XDMCP and broadcasts BroadcastQuery packets to the network. The first responding display manager will be chosen for the session.
- multicast [*address* [*hop count*]]**  
Enable XDMCP and multicast BroadcastQuery packets to the network. The first responding display manager is chosen for the session. If an address is specified, the multicast is sent to that address. If no address is specified, the multicast is sent to the default XDMCP IPv6 multicast group. If a hop count is specified, it is used as the maximum hop count for the multicast. If no hop count is specified, the multicast is set to a maximum of 1 hop, to prevent the multicast from being routed beyond the local network.
- indirect *hostname***  
enables XDMCP and send IndirectQuery packets to the specified *hostname*.
- port *port-number***  
uses the specified *port-number* for XDMCP packets, instead of the default. This option must be specified before any -query, -broadcast, -multicast, or -indirect options.

- from** *local-address*  
specifies the local address to connect from (useful if the connecting host has multiple network interfaces). The *local-address* may be expressed in any form acceptable to the host platform's `gethostbyname(3)` implementation.
- once**  
causes the server to terminate (rather than reset) when the XDMCP session ends.
- class** *display-class*  
XDMCP has an additional display qualifier used in resource lookup for display-specific options. This option sets that value, by default it is "MIT-Unspecified" (not a very useful value).
- cookie** *xdm-auth-bits*  
When testing XDM-AUTHENTICATION-1, a private key is shared between the server and the manager. This option sets the value of that private data (not that it is very private, being on the command line!).
- displayID** *display-id*  
Yet another XDMCP specific value, this one allows the display manager to identify each display so that it can locate the shared key.

## XKEYBOARD OPTIONS

X servers that support the XKEYBOARD (a.k.a. "XKB") extension accept the following options. All layout files specified on the command line must be located in the XKB base directory or a subdirectory, and specified as the relative path from the XKB base directory. The default XKB base directory is */usr/lib/X11/xkb*.

**[+–]accessx** [ *timeout* [ *timeout\_mask* [ *feedback* [ *options\_mask* ] ] ] ]  
enables(+) or disables(-) AccessX key sequences.

**–xkbdir** *directory*  
base directory for keyboard layout files. This option is not available for setuid X servers (i.e., when the X server's real and effective uids are different).

**–ardelay** *milliseconds*  
sets the autorepeat delay (length of time in milliseconds that a key must be depressed before autorepeat starts).

**–arinterval** *milliseconds*  
sets the autorepeat interval (length of time in milliseconds that should elapse between autorepeat-generated keystrokes).

**–xkbmap** *filename*  
loads keyboard description in *filename* on server startup.

## NETWORK CONNECTIONS

The X server supports client connections via a platform-dependent subset of the following transport types: TCP/IP, Unix Domain sockets, DECnet, and several varieties of SVR4 local connections. See the DISPLAY NAMES section of the X(7) manual page to learn how to specify which transport type clients should try to use.

## GRANTING ACCESS

The X server implements a platform-dependent subset of the following authorization protocols: MIT-MAGIC-COOKIE-1, XDM-AUTHORIZATION-1, XDM-AUTHORIZATION-2, SUN-DES-1, and MIT-KERBEROS-5. See the Xsecurity(7) manual page for information on the operation of these protocols.

Authorization data required by the above protocols is passed to the server in a private file named with the **–auth** command line option. Each time the server is about to accept the first connection after a reset (or when the server is starting), it reads this file. If this file contains any authorization records, the local host is not automatically allowed access to the server, and only clients which send one of the authorization records contained in the file in the connection setup information will be allowed access. See the *Xau* manual page for a description of the binary format of this file. See [xauth\(1\)](#) for maintenance of this file, and distribution of its contents to remote hosts.

The X server also uses a host-based access control list for deciding whether or not to accept connections from clients on a particular machine. If no other authorization mechanism is being used, this list initially consists of the host on which the server is running as well as any machines listed in the file */etc/Xn.hosts*, where *n* is the display number of the server. Each line of the file should contain either an Internet hostname (e.g. *expo.lcs.mit.edu*) or a DECnet hostname in double colon format (e.g. *hydra::*) or a complete name in the format *family:name* as described in the *xhost(1)* manual page. There should be no leading or trailing spaces on any lines. For example:

```
joesworkstation
corporate.company.com
star::
inet:bigcpu
local:
```

Users can add or remove hosts from this list and enable or disable access control using the *xhost* command from the same machine as the server.

If the X FireWall Proxy (*xfwp*) is being used without a *sitepolicy*, host-based authorization must be turned on for clients to be able to connect to the X server via the *xfwp*. If *xfwp* is run without a configuration file and thus no *sitepolicy* is defined, if *xfwp* is using an X server where *xhost +* has been run to turn off host-based authorization checks, when a client tries to connect to this X server via *xfwp*, the X server will deny the connection. See *xfwp(1)* for more information about this proxy.

The X protocol intrinsically does not have any notion of window operation permissions or place any restrictions on what a client can do; if a program can connect to a display, it has full run of the screen. X servers that support the SECURITY extension fare better because clients can be designated untrusted via the authorization they use to connect; see the [xauth\(1\)](#) manual page for details. Restrictions are imposed on untrusted clients that curtail the mischief they can do. See the SECURITY extension specification for a complete list of these restrictions.

Sites that have better authentication and authorization systems might wish to make use of the hooks in the libraries and the server to provide additional security models.

## SIGNALS

The X server attaches special meaning to the following signals:

### *SIGHUP*

This signal causes the server to close all existing connections, free all resources, and restore all defaults. It is sent by the display manager whenever the main user's main application (usually an *xterm* or window manager) exits to force the server to clean up and prepare for the next user.

### *SIGTERM*

This signal causes the server to exit cleanly.

### *SIGUSR1*

This signal is used quite differently from either of the above. When the server starts, it checks to see if it has inherited SIGUSR1 as SIG\_IGN instead of the usual SIG\_DFL. In this case, the server sends a SIGUSR1 to its parent process after it has set up the various connection schemes. *Xdm* uses this feature to recognize when connecting to the server is possible.

## FONTS

The X server can obtain fonts from directories and/or from font servers. The list of directories and font servers the X server uses when trying to open a font is controlled by the *font path*.

The default font path is *catalogue:/etc/X11/fontpath.d*.

A special kind of directory can be specified using the **catalogue:** prefix. Directories specified this way can contain symlinks pointing to the real font directories. See the FONTPATH.D section for details.

The font path can be set with the **-fp option or by xset(1) after the server has started.**

## FONTPATH.D

You can specify a special kind of font path in the form **catalogue:<dir>**. The directory specified after the catalogue: prefix will be scanned for symlinks and each symlink destination will be added as a local fontfile FPE.

The symlink can be suffixed by attributes such as **'unscaled'**, which will be passed through to the underlying fontfile FPE. The only exception is the newly introduced **'pri'** attribute, which will be used for ordering the font paths specified by the symlinks.

An example configuration:

```
75dpi:unscaled:pri=20 -> /usr/share/X11/fonts/75dpi
ghostscript:pri=60 -> /usr/share/fonts/default/ghostscript
misc:unscaled:pri=10 -> /usr/share/X11/fonts/misc
type1:pri=40 -> /usr/share/X11/fonts/Type1
type1:pri=50 -> /usr/share/fonts/default/Type1
```

This will add /usr/share/X11/fonts/misc as the first FPE with the attribute 'unscaled', second FPE will be /usr/share/X11/fonts/75dpi, also with the attribute 'unscaled' etc. This is functionally equivalent to setting the following font path:

```
/usr/share/X11/fonts/misc:unscaled,
/usr/share/X11/fonts/75dpi:unscaled,
/usr/share/X11/fonts/Type1,
/usr/share/fonts/default/Type1,
/usr/share/fonts/default/ghostscript
```

## FILES

<code>/etc/Xn.hosts</code>	Initial access control list for display number <b>n</b>
<code>/usr/share/fonts/X11/misc,/usr/share/fonts/X11/75dpi,/usr/share/fonts/X11/100dpi</code>	Bitmap font directories
<code>/usr/share/fonts/X11/TTF,/usr/share/fonts/X11/Type1</code>	Outline font directories
<code>/tmp/.X11-unix/Xn</code>	Unix domain socket for display number <b>n</b>
<code>/usr/adm/Xnmsgs</code>	Error log file for display number <b>n</b> if run from <a href="#">init(8)</a>
<code>/usr/lib/X11/xdm/xdm-errors</code>	Default error log file if the server is run from <code>xdm(1)</code>

## SEE ALSO

General information: [X\(7\)](#)

Protocols: *X Window System Protocol*, *The X Font Service Protocol*, *X Display Manager Control Protocol*

Fonts: [bdf2pcf\(1\)](#), [mkfontdir\(1\)](#), [mkfontscale\(1\)](#), [xfs\(1\)](#), [xlsfonts\(1\)](#), [xfontsel\(1\)](#), [xfd\(1\)](#), *X Logical Font Description Conventions*

Keyboards: [xkeyboard-config\(7\)](#)

Security: [Xsecurity\(7\)](#), [xauth\(1\)](#), [Xau\(1\)](#), [xdm\(1\)](#), [xhost\(1\)](#), [xfwp\(1\)](#), *Security Extension Specification*

Starting the server: [startx\(1\)](#), [xdm\(1\)](#), [xinit\(1\)](#)

Controlling the server once started: [xset\(1\)](#), [xsetroot\(1\)](#), [xhost\(1\)](#), [xinput\(1\)](#), [xrandr\(1\)](#)

Server-specific man pages: [Xorg\(1\)](#), [Xdmx\(1\)](#), [Xephyr\(1\)](#), [Xnest\(1\)](#), [Xvfb\(1\)](#), [Xquartz\(1\)](#), [XWin\(1\)](#).

Server internal documentation: *Definition of the Porting Layer for the X v11 Sample Server*

**AUTHORS**

The sample server was originally written by Susan Angebrandt, Raymond Drewry, Philip Karlton, and Todd Newman, from Digital Equipment Corporation, with support from a large cast. It has since been extensively rewritten by Keith Packard and Bob Scheifler, from MIT. Dave Wiggins took over post-R5 and made substantial improvements.