

NAME

X - A network transparent window system for Unix

DESCRIPTION

X is a network transparent windowing system developed at MIT which runs under Ultrix-32 Version 1.2 and 4.3BSD Unix.

X display servers run on computers with bitmap terminals. The server distributes user input to, and accepts output requests from various client programs located either on the same machine or elsewhere in the Internet. While a client normally runs on the same machine as the X server it is talking to, this need not be the case.

X supports overlapping windows, fully recursive subwindows, text and graphics operations within windows. For a full explanation of functions, see “Xlib - C Language X Interface” document.

When you first log in on a display running X, you are using the *xterm(1)* terminal emulator program. You need not learn anything extra to use a display running X as a terminal beyond moving the mouse cursor into the login window to log in normally.

X attempts to provide hooks for your favorite style of user interface; feel free to write your own if you don't like the style provided by existing window managers (see *xwm(1)*, *xnwm(1)*, or *uwm(1)*). These programs are used to manipulate existing top level windows, including moving, resizing, and iconifying existing windows. You should start your favorite window manager when you log in on a display running X.

Current client programs of X include a terminal emulator (*xterm(1)*), window managers (*xwm(1)*, *xnwm(1)* and *uwm(1)*), bitmap editor (*bitmap(1)*), access control program (*xhost(1)*), user preference setting program (*xset(1)*), load monitor (*xload(1)*), clock (*xclock(1)*), impress previewer (*ximprv(1)*), font displayer (*xfd(1)*), demos (*xdemo(1)*), and editors (e.g., *xted*). On some systems, mail notification has been integrated (*biff(1)*).

OPTIONS

The following options can be given on the command line to the X server, usually started by *init(1)* using information stored in the file */etc/tty*s. (see *ttys(5)*, *X(8c)* for details):

- a #** sets mouse acceleration (pixels)
- c** turns off key-click
- c #** sets key-click volume (0-8)
- f #** sets feep(bell) volume (0-7)
- l** sets LockUpDownMode
- l** sets LockToggleMode
- m** forces "monochrome" mode on a color display
- p #** sets screen-saver pattern cycle time (minutes)
- r** turns off auto-repeat
- r** turns on auto-repeat
- s #** sets screen-saver timeout (minutes)
- t #** sets mouse threshold (pixels)
- v** sets video-on screen-saver preference
- v** sets video-off screen-saver preference
- 0 color** sets color map entry 0 (BlackPixel)
- 1 color** sets color map entry 1 (WhitePixel)
- D rgbdb** sets RGB database file

The defaults are “-a 4 c 6 -f 3 l -p 60 r -s 10 -t 2 -0 #008 -1 #ffffff -D /usr/lib/rgb”.

X DEFAULTS

Many X programs follow the convention of using a file called *.Xdefaults* in your home directory to allow tailoring the default values of many items on the display (default font, border width, icon behavior, and so on). The format of this file is “programname.keyword:value”, where the default value for each keyword is set to the specified string. If the program name is missing, the default “keyword” value is set to the value for all programs. Case is not significant in keywords. Any whitespace before

the value is ignored. Any global defaults should precede program defaults in the file. See the manual pages for a list of what defaults can be set in a given program. Here is an overblown example `~/Xdefaults` file.

```

.BorderWidth:2
.BitmapIcon: on
.MakeWindow.Background: #8e8
.MakeWindow.Border: #f26
.MakeWindow.BodyFont: cor
.MakeWindow.Foreground: medium slate blue
.MakeWindow.Freeze: on
.MakeWindow.Mouse: #e6f
.MakeWindow.MouseMask: black
.MakeWindow.ClipToScreen: on
.Menufreeze: on
.Menubackground: maroon
.Panefont: 8x13
.SelectionFont: 8x13
.SelectionBorder: black
.Paneborderwidth: 1
xterm.Panespread: .25
biff.Background: violet red
biff.BodyFont: 9x15
biff.Border: black
biff.Foreground: green yellow
biff.Mouse: coral
bitmap.Background: forest green
bitmap.Border: salmon
bitmap.Foreground: white
bitmap.Highlight: red
bitmap.Mouse: black
xclock.Background: plum
xclock.Border: black
xclock.Foreground: red
xclock.Highlight: blue
xclock.Mode: analog
xshell.action.LeftButton: xterm =80x65-0+0 -fn 6x10
xshell.action.MiddleButton: xted =80x65+0-0
xshell.action.RightButton: xterm =20x20-0-0 -fn 6x10 -e dc
xshell.action.$: xterm =80x65+0+0 -fn 6x10 -e sh
xshell.action.#: xterm =80x65+0+0 -fn 6x10 -e su
xshell.ReverseVideo: on
xshell.WindowGeometry: =-0-0
xshell.Quiet: on
xdemo.Background: white
xdemo.Border: black
xdemo.balls.Background: maroon
xdemo.balls.Foreground: white
xdemo.circles.Foreground: khaki
xdemo.draw.Background: light gray
xdemo.draw.BodyFont: oldeng
xdemo.draw.Foreground: midnight blue
xdemo.draw.Mouse: white
xdemo.menulife.Background: medium turquoise
xdemo.menulife.Foreground: orange red
xdemo.menulife.MenuBackground: light blue
xdemo.menulife.MenuFont: oldeng
xdemo.menulife.MenuForeground: dark orchid
xdemo.menulife.MenuMouse: orange

```

<code>xdemo.menulife.Mouse:</code>	salmon
<code>xdemo.plaid.Foreground:</code>	red
<code>xdemo.qix.Foreground:</code>	violet red
<code>xdemo.slide.Foreground:</code>	forest green
<code>xdemo.wallpaper.Foreground:</code>	medium turquoise
<code>xdemo.xor.Foreground:</code>	blue violet
<code>ximpv.Background:</code>	dark green
<code>ximpv.Border:</code>	red
<code>ximpv.Foreground:</code>	cyan
<code>ximpv.Mouse:</code>	white
<code>xload.Background:</code>	<code>#ff0068</code>
<code>xload.Border:</code>	black
<code>xload.Foreground:</code>	slate blue
<code>xload.Highlight:</code>	yellow
<code>xload.ReverseVideo:</code>	on
<code>xted.Background:</code>	firebrick
<code>xted.BodyFont:</code>	kiltercrn
<code>xted.Border:</code>	tan
<code>xted.Cursor:</code>	yellow
<code>xted.Foreground:</code>	white
<code>xted.Highlight:</code>	goldenrod
<code>xted.Mouse:</code>	cyan
<code>xterm.Background:</code>	<code>#355</code>
<code>xterm.BodyFont:</code>	6x13p
<code>xterm.Cursor:</code>	green
<code>xterm.Foreground:</code>	white
<code>xfax.Background:</code>	white
<code>xfax.Border:</code>	green
<code>xfax.Foreground:</code>	red
<code>xfax.Mouse:</code>	blue

By default when you log in, only programs running on your local computer will be allowed to interact with your display. If someone else on a different machine wants to show you something, you can use the `xhost(1)` program to allow access to your display.

SIZING WINDOWS

Many programs ask you to manually size their top-level window. When started, such a program will typically popup an identification window in the upper left corner of the display. The window can be created with the center button: press the button to define one corner of the window, move the cursor to where the opposite corner of the window should be and release the button. For text applications, the left and right buttons can also be used. Pressing the left button typically produces an 80 by 24 window, which can then be moved around, and placed by releasing the button. Similarly, the right button typically produces an 80 by full screen window. For graphics applications, the left button typically creates a default size window in a default location, while the right button creates a default size window at the position of the cursor.

Most applications (e.g., `xtd`, `xdemo`, and `xfax`) read options to control sizing of initial windows. The “`MakeWindow.BodyFont`” option controls the font for the popup window. The “`MakeWindow.BorderWidth`” and “`MakeWindow.InternalBorder`” options control the outer and inner borders. The “`MakeWindow.ReverseVideo`” option can be set to “on” to reverse colors. On color displays, the “`MakeWindow.Foreground`”, “`MakeWindow.Background`”, and “`MakeWindow.Border`” options control the color of the popup window, and the “`MakeWindow.Mouse`” and “`MakeWindow.MouseMask`” options control the color of the mouse cursor. The “`MakeWindow.Freeze`” option, when set to “on”, will stop all other output while the window is sized, and use a steady outline instead of continuously flashing the window outline. The “`MakeWindow.ClipToScreen`” option will clip the resulting window to fit on the screen. (Currently only implemented in programs using the `XCreateTerm` subroutine.)

GEOMETRY SPECIFICATION

Most programs accept a geometry specification. This allows automatic creation and placement of windows on the screen at login and other convenient times. `=}XOFF}YOFF]` The `[]`'s denote optional parameters, the `{}`'s surround alternatives. WIDTH and HEIGHT are in number of characters for text

oriented applications, and usually in pixels for graphics oriented applications. XOFF and YOFF are in pixels. If you don't give XOFF and/or YOFF, then you must use the mouse to create the window. If you give XOFF and/or YOFF, then a WIDTHxHEIGHT window will automatically be creating without intervention. XOFF and YOFF specify deltas from a corner of the screen to the corresponding corner of the window, as follows:

+XOFF+YOFF	upper left to upper left
-XOFF+YOFF	upper right to upper right
+XOFF-YOFF	lower left to lower left
-XOFF-YOFF	lower right to lower right

KEYBOARD

If you don't like the standard keyboard layout or the default definitions of keymap and function keys, the keyboards on most displays can be remapped to suit your taste. Many programs look for a file called *.Xkeymap* in your home directory. This is a binary file, produced from a source map with the *keycomp(1)* program.

COLORS

Many programs allow you to specify colors for background, border, text, etc. A color specification can be given either as an english name (see */usr/lib/rgb.txt* for defined names), or three hexadecimal values for the red, green, and blue components, in one of the following formats:

```
#RGB
#RRGGBB
#RRRGGBBB
#RRRRGGGGBBBB
```

DISPLAY SPECIFICATION

When you first log in, the environment variable "DISPLAY" will be set to a string "machine:display" (for example, "mit-athena:0") which will determine which display an X application will talk to by default.

Most applications will also interpret an argument with a ":" in it to be the display to use.

When using DECnet, the format "node::display" should be used.

MENU DEFAULTS

As there is now a standard menu package for X (*XMenu(3x)*), you can tune the behavior of menus in programs using this package with a set of *Xdefaults*. *Xterm*'s 'Mode Menu' is controlled by these defaults for example.

MenuFreeze

Determines whether or not to grab the X server while a menu is posted. One of: on, off. The default value is off.

MenuStyle

Determines the menu display style. One of: left_hand, right_hand, center. The default value is right_hand.

MenuMode

Determines the menu selection high light mode. One of: box, invert. If box mode is chosen then the SelectionBorderWidth and SelectionBorderColor parameters effect the box line width and color respectively. If invert mode is chose then the SelectionForeground and MenuBackground colors are used for the inversion. The default value is invert.

MenuMouse

Determines the color of the mouse cursor while it is within the menu. Any valid X color may be used. The default value is black.

MenuBackground

Determines the menu background color. Any valid X color may be used. The default value is white.

MenuInactivePattern

Determines which of the five possible bitmap patterns will be used to tile inactive panes. One of: `dimple1`, `dimple3`, `gray1`, `gray3`, `cross_weave`. The default value is `gray3`.

PaneStyle

Determines the display style of all menu panes. One of: `flush_left`, `flush_right`, `center`. The default value is `center`.

PaneFont

Determines the font used for the label (heading text) of each pane. Any valid X font may be used. The default value is `8x13`.

PaneForeground

Determines the pane foreground color. This is the color used for the label (heading text) in each pane. Any valid X color may be used. The default value is black.

PaneBorder

Determines the color of all menu pane borders. Any valid X color may be used. The default value is black.

PaneBorderWidth

Determines the width (in pixels) of all menu pane borders. Any integer greater than or equal to 0 may be used. The default value is 2.

PaneSpread

Determines the horizontal spread of menu panes. Any double greater than or equal to 0.0 may be used. A value of 1.0 specifies a one to one ratio between horizontal spread and vertical spread. A value less than 1.0 will compress the menu panes inward and a value greater than 1.0 will expand them outward. The default value is 1.0.

SelectionStyle

Determines the display style of all menu selections. One of: `flush_left`, `flush_right`, `center`. The default value is `flush_left`.

SelectionFont

Determines the font used for the text in each selection. Any valid X font may be used. The default value is `6x10`.

SelectionForeground

Determines the selection foreground color. This is the color used for the text in each selection. Any valid X color may be used. The default value is black.

SelectionBorder

Determines the color of all menu selection borders. Any valid X color may be used. The default value is black.

SelectionBorderWidth

Determines the width (in pixels) of all menu selection borders. Any integer greater than or equal to 0 may be used. The default value is 1.

SelectionSpread

Determines the inter-selection spread. Any double greater than or equal to 0.0 may be used. A value of 1.0 specifies that 1.0 times the height of the current selection font will be used for padding. The default value is 0.25.

SEE ALSO

`X(8c)`, `xterm(1)`, `bitmap(1)`, `xwm(1)`, `xnwm(1)`, `xhost(1)`, `xclock(1)`, `xload(1)`, `xset(1)`, `keycomp(1)`, `xdemo(1)`, `biff(1)`, `qv(4)`, `vs(4)`, `init(8)`, `ttys(5)`, `uwm(1)`, `xrefresh(1)`, `xwininfo(1)`, `ximpv(1)`, `xdvi(1)`, `pikapix(1)`, `xwd(1)`, `xwud(1)`, `xinit(1)`, `xtd(1)`, `xdemo(1)`, `Xqvss(8c)`, `Xvs100(8c)`, `Xsun(8c)`, `Xnest(8c)`
‘Xlib - C Language X Interface’

AUTHORS

It is no longer feasible to list all people who have contributed something to X; below is a short list of people who have added significant code to device independent parts of X.

Bob Scheifler (MIT-LCS), Jim Gettys (MIT-Project Athena, DEC), Mark Vandevoorde (MIT-Project Athena, DEC), Tony Della Fera (MIT-Project Athena, DEC), Ron Newman (MIT-Project Athena, MIT), Shane Hartman and Stuart Malone (MIT-LCS), Doug Mink (Smithsonian Astrophysical Observatory), Bob McNamara (DEC-MAD), and Stephen Sutphen (University of Alberta).

Special thanks must go to Paul Asente (of DECWRL and Stanford University), who wrote "W" which saved us much time and energy early in this project, and who is now an active X contributor as well, and Chris Kent (of DECWRL and Purdue University) who both struggled mightily (and won!) to turn the Vs100 into something useful under Unix.

We are very grateful for the interest shown by many groups in the country, which has encouraged us to make X more than our personal toy. Great thanks must go to Digital's Ultrix Engineering Group for the QDSS implementation, and to Digital's Workstations Group for the QVSS implementation.

Copyright (c) 1984, 1985, 1986 by Massachusetts Institute of Technology.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

This software is not subject to any license of the American Telephone and Telegraph Company or of the Regents of the University of California.