

**NAME**

*xwm* - X Window System, window manager process

**SYNOPSIS**

**xwm** [ *-cfgmrsz* ] [ *+function* ] [ *@delta* ] [ *fn=font* ] [ *fi=font* ] [ *host:display* ]

**DESCRIPTION**

The window manager allows you to use the mouse to push a window to the top or bottom of the stack, turn a window into an icon, resize a window, move a window elsewhere on the screen, attach the keyboard to a window (hierarchy) and circulate the window hierarchy. The window manager only manipulates top-level windows (i.e., direct decedents of the root window), not their subwindows, so in the following, references to window refer only to top-level windows.

Since *xwm* does not have a window of its own it steals certain button/key combinations and interprets them as operations on existing windows. The key combination is specified on the command line with some subset of the options: 'c' (control), 's' (shift), 'm' (meta) and 'n' (no-key). For example, if you specify the options **-cm** then the Control and Meta keys must be down at the time a mouse button is depressed. If no combination is specified in the command line, Meta is the default. If 'n' is specified anywhere in the option list all keys will be ignored.

For each mouse button, a different command is performed depending on whether the button is 'clicked' or 'moved', i.e., whether the mouse is moved between the press and release of the button. Some actual movement is allowed before the mouse is really considered to have moved, the amount of movement is settable (see below). The mouse buttons perform the commands described below. For any of these commands, if you press a button to start a command, and then want to abort the command, simply press one of the other buttons before releasing the first button. As each command is being performed the mouse cursor will be changed to indicate which command is in effect.

If the left button is clicked in a window it will 'push' the window you are pointing at to the bottom of any stack of overlapping windows. If clicked on the root window a 'circulate down' operation will be performed on the root window moving the top most window in the hierarchy to the bottom. For any of these operations the mouse cursor will be a 'dot'.

The left button will also 'iconify' the window you point at if it is pressed down and then moved. When you release the button, the window will be made into an icon at the current mouse location. If the window being iconified has its own icon, then that icon will be used. If not *xwm* will create and maintain its own text icon using the name of the window as the initial text. For any of these operations the mouse cursor will be an 'icon' cursor.

The name displayed in an *xwm* owned text icon can be edited at any time by placing the mouse cursor in the icon and typing. Note: Modifying text displayed in an icon window will modify the name of the window associated with that icon. The delete key deletes the last character, Control-U deletes the entire name, any other printing characters are appended to the current name. When the mouse cursor in an *xwm* text icon it will be a 'text' cursor ('I bar' cursor).

If you click the middle button on an icon, the window you iconified will reappear in its previous position on the screen and the icon will disappear. For this operation the mouse cursor will be an 'arrow cross' cursor.

The middle button is used to resize a window by moving a corner or an edge. If you press it on a window, a rubber banded outline of the window will be displayed (and a grid if you specify the 'g' option explained below) and moving the mouse will change its size, leaving the opposite corner or other edges fixed. The corner or edge to be moved depends on the where the mouse is when the button is pressed. Imagine the window divided with grid of nine rectangles (the same grid that the 'g' option displays). If the mouse is in one of the four corner rectangles or the center rectangle, then the corner closest to the mouse will be moved; otherwise, the closest edge will be moved. When the button is released, the window will be resized. For these operations the mouse cursor will be an 'arrow cross' cursor.

The middle button can also be used to focus keyboard input to a specific window i.e., keyboard input will go to the specified window (hierarchy) even when the mouse is outside the window. If the 'f' option is specified clicking the middle button twice on a window will attach the keyboard to that

window. Clicking the middle button on the background will detach the keyboard from any window (actually, it attaches it to the background window). For this operation the mouse cursor will be an 'arrow cross' cursor.

The right button, if clicked in a window, will 'pull' the window you are pointing at to the top of any stack of overlapping windows. If clicked on the root window a 'circulate up' operation will be performed moving the bottom most window in the hierarchy to the top. For these operations the mouse cursor will be a 'circle' cursor.

The right button will also move the window you are pointing at if it is pressed down and then moved. An outline of the window (and a grid if you specified the 'g' option) will appear, and will move with the mouse cursor. When you release the right button, the window will be moved to the current location of the outline. For this operation the mouse cursor will be a 'circle' cursor.

#### OPTION SUMMARY:

- c** The 'c' (control) option specifies that the Control key must be held down for *xwm* to listen to mouse button operations.
- f** The 'f' (focus) option specifies that a double-click on the middle button will focus keyboard input events to the specified window.
- g** The (grid) option turns on a tick-tack-toe like grid that will be displayed inside the 'window box' during window movement and resize operations.
- m** The 'm' (meta) option specifies that the Meta key must be held down for *xwm* to listen to mouse button operations.
- n** The 'n' (no-key) option specifies no keys may be down when performing mouse button operation.
- r** The 'r' (reverse) option indicates that icons should be displayed as white text on a black background, rather than black text on a white background.
- s** The 's' (shift) options indicates that the Shift key must be held down for *xwm* to listen to mouse button operations.
- z** The 'z' (zap) option turns on a special 'zap' effect that is intended to draw your attention to icons as they are created and windows as they are moved.

#### ARGUMENT SUMMARY:

*+function*

This argument allows you to specify a cursor display function. It should be followed by an integer specifying the code of the display function. See the Xlib document for details of available functions. The default function is GXcopy.

**@delta** This argument allows you to specify a mouse *delta* value. This value determines how far the mouse must move with a button down before the iconify, move and change operations begin. The default is 5 pixels. Note that if you define a large delta, you can still make fine adjustments by first moving far away and then moving back.

**fn=font** This argument allow you to specify a text *font* to be used in pop up information display. The default font is 6x10.

**fi=font** This argument allow you to specify an icon text font. The default font is 6x10.

*host:display*

This argument allow you to specify the the host and display number on which *xwm* will operate. For example 'xwm orpheus:1' would start up the window manager on display one on the machine orpheus. By default, *xwm* uses the host and display number stored in the environment variable DISPLAY, and therefore this argument is not normally specified. The window manager can be running anywhere, and you can run more than one manager per display (provided that they do not attempt to use the same mouse button / key combinations, see below).

**X DEFAULTS****BodyFont**

Set the default font for information display.

**IconFont**

Set the default font for text icons.

**InternalBorder**

Set the space between the text and window border in text icons.

**BorderWidth**

Set the border width of text icons.

**ReverseVideo**

Display text icons in reverse video?

**ENVIRONMENT****DISPLAY**

To get default host and display number.

**SEE ALSO**

X(1), X(8C)

**AUTHOR**

Copyright 1985, Massachusetts Institute of Technology.

Tony Della Fera, DEC MIT Project Athena

Based upon previous 'xwm' by Bob Scheifler, MIT Laboratory for Computer Science