NAME

XSetWMName, XGetWMName, XStoreName, XFetchName - set or read a window’s WM_NAME property

SYNTAX

void XSetWMName(Display **display*, Window *w*, XTextProperty **text_prop*);

Status XGetWMName(Display **display*, Window *w*, XTextProperty **text_prop_return*);

int XStoreName(Display **display*, Window *w*, _Xconst char **window_name*);

Status XFetchName(Display **display*, Window *w*, char ***window_name_return*);

ARGUMENTS

display Specifies the connection to the X server.

text_prop Specifies the **XTextProperty** structure to be used.

text_prop_return
Returns the **XTextProperty** structure.

w Specifies the window.

window_name Specifies the window name, which should be a null-terminated string.

window_name_return
Returns the window name, which is a null-terminated string.

DESCRIPTION

The **XSetWMName** convenience function calls **XSetTextProperty** to set the WM_NAME property.

The **XGetWMName** convenience function calls **XGetTextProperty** to obtain the WM_NAME property. It returns a nonzero status on success; otherwise, it returns a zero status.

The **XStoreName** function assigns the name passed to window_name to the specified window. A window manager can display the window name in some prominent place, such as the title bar, to allow users to identify windows easily. Some window managers may display a window’s name in the window’s icon, although they are encouraged to use the window’s icon name if one is provided by the application. If the string is not in the Host Portable Character Encoding, the result is implementation-dependent.

XStoreName can generate **BadAlloc** and **BadWindow** errors.

The **XFetchName** function returns the name of the specified window. If it succeeds, it returns a nonzero status; otherwise, no name has been set for the window, and it returns zero. If the WM_NAME property has not been set for this window, **XFetchName** sets window_name_return to NULL. If the data returned by the server is in the Latin Portable Character Encoding, then the returned string is in the Host Portable Character Encoding. Otherwise, the result is implementation-dependent. When finished with it, a client must free the window name string using **XFree**.

XFetchName can generate a **BadWindow** error.

PROPERTIES

WM_NAME The name of the application.

DIAGNOSTICS

BadAlloc The server failed to allocate the requested resource or server memory.

BadWindow A value for a Window argument does not name a defined Window.

SEE ALSO

XAllocClassHint(3), XAllocIconSize(3), XAllocSizeHints(3), XAllocWMHints(3), XFree(3), XSetCommand(3), XSetTransientForHint(3), XSetTextProperty(3), XSetWMClientMachine(3), XSetWMColormapWindows(3), XSetWMIconName(3), XSetWMProperties(3), XSetWMProtocols(3), XStringListToTextProperty(3)
Xlib - C Language X Interface