

**NAME**

XAllocClassHint, XSetClassHint, XGetClassHint, XClassHint - allocate class hints structure and set or read a window's WM\_CLASS property

**SYNTAX**

```
XClassHint *XAllocClassHint(void);
```

```
XSetClassHint(Display *display, Window w, XClassHint *class_hints);
```

```
Status XGetClassHint(Display *display, Window w, XClassHint *class_hints_return);
```

**ARGUMENTS**

*display* Specifies the connection to the X server.

*class\_hints* Specifies the **XClassHint** structure that is to be used.

*class\_hints\_return*  
Returns the **XClassHint** structure.

*w* Specifies the window.

**DESCRIPTION**

The **XAllocClassHint** function allocates and returns a pointer to a **XClassHint** structure. Note that the pointer fields in the **XClassHint** structure are initially set to NULL. If insufficient memory is available, **XAllocClassHint** returns NULL. To free the memory allocated to this structure, use **XFree**.

The **XSetClassHint** function sets the class hint for the specified window. If the strings are not in the Host Portable Character Encoding, the result is implementation-dependent.

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

The **XGetClassHint** function returns the class hint of the specified window to the members of the supplied structure. If the data returned by the server is in the Latin Portable Character Encoding, then the returned strings are in the Host Portable Character Encoding. Otherwise, the result is implementation-dependent. It returns a nonzero status on success; otherwise, it returns a zero status. To free *res\_name* and *res\_class* when finished with the strings, use **XFree** on each individually.

**XGetClassHint** can generate a **BadWindow** error.

**PROPERTIES**

**WM\_CLASS** Set by application programs to allow window and session managers to obtain the application's resources from the resource database.

## STRUCTURES

The **XClassHint** structure contains:

```
typedef struct {
    char *res_name;
    char *res_class;
} XClassHint;
```

The `res_name` member contains the application name, and the `res_class` member contains the application class.

Note that the name set in this property may differ from the name set as `WM_NAME`. That is, `WM_NAME` specifies what should be displayed in the title bar and, therefore, can contain temporal information (for example, the name of a file currently in an editor's buffer).

On the other hand, the name specified as part of `WM_CLASS` is the formal name of the application that should be used when retrieving the application's resources from the resource database.

## 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

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