XSetTextProperty, XGetTextProperty – set and read text properties

```
void XSetTextProperty(display, w, text_prop, property)
      Display *display;
      Window w;
      XTextProperty *text_prop;
      Atom property;
```

```
Status XGetTextProperty(display, w, text_prop_return, property)
       Display *display;
       Window w;
       XTextProperty *text_prop_return;
       Atom property;
```

display Specifies the connection to the X server.

property Specifies the property name.

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

text_prop_return Returns the **XTextProperty** structure.

The XSetTextProperty function replaces the existing specified property for the named window with the data, type, format, and number of items determined by the value field, the encoding field, the format field, and the nitems field, respectively, of the specified **XTextProperty** structure. If the property does not already exist, **XSetTextProperty** sets it for the specified window.

XSetTextProperty can generate **BadAlloc**, **BadAtom**, **BadValue**, and **BadWindow** errors.

The **XGetTextProperty** function reads the specified property from the window and stores the data in the returned **XTextProperty** structure. It stores the data in the value field, the type of the data in the encoding field, the format of the data in the format field, and the number of items of data in the nitems field. An extra byte containing null (which is not included in the nitems member) is stored at the end of the value field of text_prop_return. The particular interpretation of the property's encoding and data as text is left to the calling application. If the specified property does not exist on the window, **XGetTextProperty** sets the value field to NULL, the encoding field to **None**, the format field to zero, and the nitems field to zero.

If it was able to read and store the data in the **XTextProperty** structure, **XGetTextProperty** returns a nonzero status; otherwise, it returns a zero status.

XGetTextProperty can generate **BadAtom** and **BadWindow** errors.

WM_CLIENT_MACHINE The string name of the machine on which the client application is running. WM_COMMAND The command and arguments, null-separated, used to invoke the application. WM_ICON_NAME The name to be used in an icon. WM_NAME The name of the application.

BadAlloc The server failed to allocate the requested resource or server memory. **BadAtom** A value for an Atom argument does not name a defined Atom. **BadValue** Some numeric value falls outside the range of values accepted by the request. Unless a specific range is specified for an argument, the full range defined by the argument's type is accepted. Any argument defined as a set of alternatives can generate this error. **BadWindow** A value for a Window argument does not name a defined Window.

XAllocClassHint(3X11), XAllocIconSize(3X11), XAllocSizeHints(3X11), XAllocWMHints(3X11), XSetCommand(3X11), XSetTransientForHint(3X11), XSetWMClientMachine(3X11), XSetWMColormapWindows(3X11), XSetWMIconName(3X11), XSetWMName(3X11), XSetWMProperties(3X11), XSetWMProtocols(3X11), XStringListToTextProperty(3X11)
Xlib – C Language X Interface