XGetWindowProperty, XListProperties, XChangeProperty, XRotateWindowProperties, XDeleteProperty – obtain and change window properties

```
int XGetWindowProperty(display, w, property, long_offset, long_length, delete, req_type,
                    actual_type_return, actual_format_return, nitems_return, bytes_after_return,
                    prop_return)
     Display *display;
     Window w;
     Atom property;
     long long_offset, long_length;
     Bool delete;
     Atom req_type;
     Atom *actual_type_return;
     int *actual_format_return;
     unsigned long *nitems_return;
     unsigned long *bytes_after_return;
     unsigned char **prop_return;

Atom *XListProperties(display, w, num_prop_return)
     Display *display;
     Window w;
     int *num_prop_return;

XChangeProperty(display, w, property, type, format, mode, data, nelements)
     Display *display;
     Window w;
     Atom property, type;
     int format;
     int mode;
     unsigned char *data;
     int nelements;

XRotateWindowProperties(display, w, properties, num_prop, npositions)
     Display *display;
     Window w;
     Atom properties[];
     int num_prop;
     int npositions;

XDeleteProperty(display, w, property)
     Display *display;
     Window w;
     Atom property;
```

actual_format_return	Returns the actual format of the property.
actual_type_return	Returns the atom identifier that defines the actual type of the property.
bytes_after_return	Returns the number of bytes remaining to be read in the property if a partial read was performed.
data	Specifies the property data.
delete	Specifies a Boolean value that determines whether the property is deleted.
display	Specifies the connection to the X server.
format	Specifies whether the data should be viewed as a list of 8-bit, 16-bit, or 32-bit quantities. Possible values are 8, 16, and 32. This information allows the X server to correctly perform byte-swap operations as necessary. If the format is 16-bit or 32-bit, you must

	explicitly cast your data pointer to an (unsigned char *) in the call to **XChangeProperty**.
long_length	Specifies the length in 32-bit multiples of the data to be retrieved.
long_offset	Specifies the offset in the specified property (in 32-bit quantities) where the data is to be retrieved.
mode	Specifies the mode of the operation. You can pass **PropModeReplace**, **PropModePrepend**, or **PropModeAppend**.
nelements	Specifies the number of elements of the specified data format.
nitems_return	Returns the actual number of 8-bit, 16-bit, or 32-bit items stored in the prop_return data.
num_prop	Specifies the length of the properties array.
num_prop_return	Returns the length of the properties array.
npositions	Specifies the rotation amount.
prop_return	Returns the data in the specified format.
property	Specifies the property name.
properties	Specifies the array of properties that are to be rotated.
req_type	Specifies the atom identifier associated with the property type or **AnyPropertyType**.
type	Specifies the type of the property. The X server does not interpret the type but simply passes it back to an application that later calls **XGetWindowProperty**.
w	Specifies the window whose property you want to obtain, change, rotate or delete.

The XGetWindowProperty function returns the actual type of the property; the actual format of the property; the number of 8-bit, 16-bit, or 32-bit items transferred; the number of bytes remaining to be read in the property; and a pointer to the data actually returned. **XGetWindowProperty** sets the return arguments as follows:

- If the specified property does not exist for the specified window, **XGetWindowProperty** returns **None** to actual_type_return and the value zero to actual_format_return and bytes_after_return. The nitems_return argument is empty. In this case, the delete argument is ignored.
- If the specified property exists but its type does not match the specified type, **XGetWindowProperty** returns the actual property type to actual_type_return, the actual property format (never zero) to actual_format_return, and the property length in bytes (even if the actual_format_return is 16 or 32) to bytes_after_return. It also ignores the delete argument. The nitems_return argument is empty.
- If the specified property exists and either you assign **AnyPropertyType** to the req_type argument or the specified type matches the actual property type, **XGetWindowProperty** returns the actual property type to actual_type_return and the actual property format (never zero) to actual_format_return. It also returns a value to bytes_after_return and nitems_return, by defining the following values:

  ```
  N = actual length of the stored property in bytes
      (even if the format is 16 or 32)
  I = 4 * long_offset
  T = N - I
  L = MINIMUM(T, 4 * long_length)
  A = N - (I + L)
  ```

  The returned value starts at byte index I in the property (indexing from zero), and its length in bytes is L. If the value for long_offset causes L to be negative, a **BadValue** error results. The value of bytes_after_return is A, giving the number of trailing unread bytes in the stored property.

If the returned format is 8, the returned data is represented as a **char** array. If the returned format is 16, the returned data is represented as a **short** array and should be cast to that type to obtain the elements. If the returned format is 32, the returned data is represented as a **long** array and should be cast to that type to obtain the elements.

XGetWindowProperty always allocates one extra byte in prop_return (even if the property is zero length) and sets it to zero so that simple properties consisting of characters do not have to be copied into yet another string before use.

If delete is **True** and bytes_after_return is zero, **XGetWindowProperty** deletes the property from the window and generates a **PropertyNotify** event on the window.

The function returns **Success** if it executes successfully. To free the resulting data, use **XFree**.

XGetWindowProperty can generate **BadAtom**, **BadValue**, and **BadWindow** errors.

The **XListProperties** function returns a pointer to an array of atom properties that are defined for the specified window or returns NULL if no properties were found. To free the memory allocated by this function, use **XFree**.

XListProperties can generate a **BadWindow** error.

The **XChangeProperty** function alters the property for the specified window and causes the X server to generate a **PropertyNotify** event on that window. **XChangeProperty** performs the following:

- If mode is **PropModeReplace**, **XChangeProperty** discards the previous property value and stores the new data.
- If mode is **PropModePrepend** or **PropModeAppend**, **XChangeProperty** inserts the specified data before the beginning of the existing data or onto the end of the existing data, respectively. The type and format must match the existing property value, or a **BadMatch** error results. If the property is undefined, it is treated as defined with the correct type and format with zero-length data.

If the specified format is 8, the property data must be a **char** array. If the specified format is 16, the property data must be a **short** array. If the specified format is 32, the property data must be a **long** array.

The lifetime of a property is not tied to the storing client. Properties remain until explicitly deleted, until the window is destroyed, or until the server resets. For a discussion of what happens when the connection to the X server is closed, see section 2.6. The maximum size of a property is server dependent and can vary dynamically depending on the amount of memory the server has available. (If there is insufficient space, a **BadAlloc** error results.)

XChangeProperty can generate **BadAlloc**, **BadAtom**, **BadMatch**, **BadValue**, and **BadWindow** errors.

The **XRotateWindowProperties** function allows you to rotate properties on a window and causes the X server to generate **PropertyNotify** events. If the property names in the properties array are viewed as being numbered starting from zero and if there are num_prop property names in the list, then the value associated with property name I becomes the value associated with property name (I + npositions) mod N for all I from zero to N – 1. The effect is to rotate the states by npositions places around the virtual ring of property names (right for positive npositions, left for negative npositions). If npositions mod N is nonzero, the X server generates a **PropertyNotify** event for each property in the order that they are listed in the array. If an atom occurs more than once in the list or no property with that name is defined for the window, a **BadMatch** error results. If a **BadAtom** or **BadMatch** error results, no properties are changed.

XRotateWindowProperties can generate **BadAtom**, **BadMatch**, and **BadWindow** errors.

The **XDeleteProperty** function deletes the specified property only if the property was defined on the specified window and causes the X server to generate a **PropertyNotify** event on the window unless the property does not exist.

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

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.

XFree(3X11), XInternAtom(3X11)
Xlib – C Language X Interface