NAME

XIChangeProperty, XIGetProperty, XIDeleteProperty - change, retrieve or delete a device’s property.

SYNOPSIS

```
#include <X11/extensions/XInput2.h>

void XIChangeProperty( Display* dpy,
                       int deviceid,
                       Atom property,
                       Atom type,
                       int format,
                       int mode,
                       unsigned char *data,
                       int num_items)

void XIDeleteProperty( Display *dpy,
                       int deviceid,
                       Atom property)

Status XIGetProperty( Display *dpy,
                      int deviceid,
                      Atom property,
                      long offset,
                      long length,
                      Bool delete_property,
                      Atom type,
                      Atom *type_return,
                      int *format_return,
                      unsigned long *num_items_return,
                      unsigned long *bytes_after_return,
                      unsigned char **data)
```

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.

display

Specifies the connection to the X server.

delete_property
: Specifies a Boolean value that determines whether the property is to be deleted.

deviceid
: The device to list the properties for.

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 XIChangeProperty.

format_return
: Returns the actual format of the property.

length
: Specifies the length in 32-bit multiples of the data to be retrieved.

offset
: Specifies the offset in the specified property (in 32-bit quantities) where the data is to be retrieved.

mode
: One of XIPropModeAppend, XIPropModePrepend or XIPropModeReplace.

num_items
: Number of items in data in the format specified.

nitems_return
: Returns the actual number of 8-bit, 16-bit, or 32-bit items stored in data.

property

Specifies the property name.

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 XIGetProperty.

type_return
Returns the atom identifier that defines the actual type of the property.

DESCRIPTION

The XIGetProperty 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. XIGetProperty sets the return arguments as follows:

- the specified property does not exist for the specified device, XIGetProperty 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.

- the specified property exists but its type does not match the specified type, XIGetProperty 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.

- the specified property exists and either you assign XIAnyPropertyType to the req_type argument or the specified type matches the actual property type, XIGetProperty 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 = length of the stored property in bytes
I = 4 * offset
T = N - I
L = MINIMUM(T, 4 * 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 uint16_t 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 uint32_t array and should be cast to that type to obtain the elements.

XIGetProperty 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, XIGetProperty deletes the property from the window and generates an XIPropertyNotify event on the window.

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

XIGetProperty can generate BadAtom, BadValue, and BadWindow errors.

The XIChangeProperty function alters the property for the specified device and causes the X server to generate a XIPropertyNotify event for that device. XIChangeProperty performs the following:

- mode is XIPropModeReplace, XIChangeProperty discards the previous property value and stores the new data.

- mode is XIPropModePrepend or XIPropModeAppend, XIChangeProperty 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 uint16_t array. If the specified format is 32, the property data must be a

uint32_t array.

The lifetime of a property is not tied to the storing client. Properties remain until explicitly deleted, until the device is removed, or until the server resets. 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.)

XIChangeProperty can generate BadAlloc, BadAtom, BadMatch, BadValue, and BadDevice errors.

The XIDeleteProperty function deletes the specified property only if the property was defined on the specified device and causes the X server to generate a XIPropertyNotify event for the device unless the property does not exist.

XIDeleteProperty can generate BadAtom and BadDevice errors.

DIAGNOSTICS

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.

BadDevice An invalid device was specified. The device does not exist.

BadAtom An invalid property was specified. The property does not exist.

BUGS

The protocol headers for XI 2.0 did not provide XIAnyPropertyType, XIPropModeReplace, XIPropModePrepend or XIPropModeAppend. Use AnyPropertyType, PropModeReplace, PropModePrepend and PropModeAppend instead, respectively.

SEE ALSO

XIListProperties(3)