

**NAME**

XIGrabEnter, XIUngrabEnter, XIGrabFocusIn, XIUngrabFocusIn - grab/ungrab enter or focus in events.

**SYNOPSIS**

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

```
int XIGrabEnter( Display *display,
                int deviceid,
                Window grab_window,
                Cursor cursor,
                int grab_mode,
                int paired_device_mode,
                Bool owner_events,
                XIEventMask *mask,
                int num_modifiers,
                XIGrabModifiers *modifiers_inout);
```

```
int XIUngrabEnter( Display *display,
                  int deviceid,
                  Window grab_window
                  int num_modifiers,
                  XIGrabModifiers *modifiers);
```

```
int XIGrabFocusIn ( Display *display,
                    int deviceid,
                    Window grab_window
                    int grab_mode,
                    int paired_device_mode,
                    Bool owner_events,
                    XIEventMask *mask,
                    int num_modifiers,
                    XIGrabModifiers *modifiers_inout);
```

```
int XIUngrabFocusIn( Display *display,
                     int deviceid,
                     Window grab_window
                     int num_modifiers,
                     XIGrabModifiers *modifiers);
```

display

Specifies the connection to the X server.

device

Specifies the device that is to be grabbed or released

num\_modifiers

Number of elements in modifiers or modifiers\_inout.

modifiers

Specifies the set of latched and base modifiers or XIAnyModifier to ungrab. The data type is for consistency with the respective grab request and the status code of the XIGrabModifiers struct is ignored.

modifiers\_inout

Specifies the set of latched and base modifiers or XIAnyModifier to grab. Returns the modifiers that could not be grabbed and their status code.

grab\_window

Specifies the grab window.

owner\_events

Specifies a Boolean value that indicates whether the are to be reported as usual or reported with respect to the grab window.

mask

Specifies the event mask.

grab\_mode

Specifies further processing of events from this device. You can pass XIGrabModeSync or XIGrabModeAsync.

paired\_device\_mode

Specifies further processing of events from the paired master device. You can pass XIGrabModeSync or XIGrabModeAsync. If deviceid specifies a floating slave device, this parameter is ignored.

**DESCRIPTION**

XIGrabEnter and XIGrabFocusIn establish a passive grab. The modifier device for a enter grab is the paired master device if deviceid specifies a master pointer. Otherwise, the modifier device is the device specified with deviceid.

For XIGrabEnter and XIGrabFocusIn, in the future, the device is actively grabbed (as for XIGrabDevice, the last-grab time is set to the time at which the pointer/focus entered window) if the device is not grabbed and the pointer has entered the grab\_window or the focus has been set to the grab\_window when the specified modifier keys are logically down on the modifier device and no other buttons or modifier keys are logically down.

The interpretation of the remaining arguments is as for XIGrabDevice. The active grab is terminated automatically when the pointer leaves the window or the focus is set to a different window (independent of the logical state of the modifier keys).

If the device is an attached slave device, the device is automatically detached from the master device when the grab activates and reattached to the same master device when the grab deactivates. If the master device is removed while the device is floating as a result of a grab, the device remains floating once the grab deactivates.

Note that the logical state of a device (as seen by client applications) may lag the physical state if device event processing is frozen.

This request overrides all previous grabs by the same client of the same type and modifier combinations on the same window. A modifiers of XIAnyModifier is equivalent to issuing the grab request for all possible modifier combinations (including the combination of no modifiers). It is not required that all modifiers specified have currently assigned KeyCodes.

If some other client has already issued a XIGrabEnter or

XIGrabFocusIn on the same window, a BadAccess error results. When using XIAnyModifier, the request fails completely, and a XIBadAccess error results (no grabs are established) if there is a conflicting grab for any combination. XIGrabEnter and XIGrabFocusIn have no effect on an active grab.

On success, XIGrabEnter and XIGrabFocusIn return 0; If one or more modifier combinations could not be grabbed, XIGrabEnter and XIGrabFocusIn return the number of failed combinations and modifiers\_inout contains the failed combinations and their respective status codes.

XIGrabEnter and XIGrabFocusIn can generate BadDevice, BadMatch, BadValue, and BadWindow errors.

XIUngrabEnter and XIUngrabFocusIn releases the passive grab on the specified window if it was grabbed by this client. A modifier of XIAnyModifier is equivalent to issuing the ungrab request for all possible modifier combinations, including the combination of no modifiers. XIUngrabEnter and XIUngrabFocusIn have no effect on an active grab.

XIUngrabEnter and XIUngrabFocusIn can generate BadDevice, BadMatch, BadValue and BadWindow errors.

## RETURN VALUE

XIGrabEnter and XIGrabFocusIn return the number of modifier combination that could not establish a passive grab. The modifiers are returned in modifiers\_inout, along with the respective error for this modifier combination. If XIGrabEnter or XIGrabFocusIn return zero, passive grabs with all requested modifier combinations were established successfully.

## DIAGNOSTICS

### BadDevice

An invalid deviceid was specified.

### BadMatch

This error may occur if XIGrabEnter specified a device that has no valuator, or XIGrabFocusIn specified a device that has no keys.

**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.

**BUGS**

The protocol headers for XI 2.0 did not provide XIGrabModeAsync or XIGrabModeSync. Use GrabModeSync and GrabModeAsync instead, respectively.

**SEE ALSO**

XIAllowEvents(3)