

NAME

`xcb_grab_button` - Grab pointer button(s)

SYNOPSIS

```
#include <xcb/xproto.h>
```

Request function

```
xcb_void_cookie_t xcb_grab_button(xcb_connection_t *conn, uint8_t owner_events,
    xcb_window_t grab_window, uint16_t event_mask, uint8_t pointer_mode,
    uint8_t keyboard_mode, xcb_window_t confine_to, xcb_cursor_t cursor, uint8_t button,
    uint16_t modifiers);
```

REQUEST ARGUMENTS

conn The XCB connection to X11.

owner_events If 1, the *grab_window* will still get the pointer events. If 0, events are not reported to the *grab_window*.

grab_window Specifies the window on which the pointer should be grabbed.

event_mask Specifies which pointer events are reported to the client.

 TODO: which values?

pointer_mode One of the following values:

XCB_GRAB_MODE_SYNC

The state of the keyboard appears to freeze: No further keyboard events are generated by the server until the grabbing client issues a releasing *AllowEvents* request or until the keyboard grab is released.

XCB_GRAB_MODE_ASYNC

Keyboard event processing continues normally.

keyboard_mode

One of the following values:

XCB_GRAB_MODE_SYNC

The state of the keyboard appears to freeze: No further keyboard events are generated by the server until the grabbing client issues a releasing *AllowEvents* request or until the keyboard grab is released.

XCB_GRAB_MODE_ASYNC

Keyboard event processing continues normally.

confine_to Specifies the window to confine the pointer in (the user will not be able to move the pointer out of that window).

The special value *XCB_NONE* means don't confine the pointer.

cursor Specifies the cursor that should be displayed or *XCB_NONE* to not change the cursor.

button One of the following values:

XCB_BUTTON_INDEX_ANY

Any of the following (or none):

XCB_BUTTON_INDEX_1

The left mouse button.

XCB_BUTTON_INDEX_2

The right mouse button.

XCB_BUTTON_INDEX_3

The middle mouse button.

XCB_BUTTON_INDEX_4

Scroll wheel. TODO: direction?

XCB_BUTTON_INDEX_5

Scroll wheel. TODO: direction?

modifiers The modifiers to grab.

Using the special value *XCB_MOD_MASK_ANY* means grab the pointer with all possible modifier combinations.

DESCRIPTION

This request establishes a passive grab. The pointer is actively grabbed as described in *GrabPointer*, the last-pointer-grab time is set to the time at which the button was pressed (as transmitted in the *ButtonPress* event), and the *ButtonPress* event is reported if all of the following conditions are true:

The pointer is not grabbed and the specified button is logically pressed when the specified modifier keys are logically down, and no other buttons or modifier keys are logically down.

The grab-window contains the pointer.

The confine-to window (if any) is viewable.

A passive grab on the same button/key combination does not exist on any ancestor of grab-window.

The interpretation of the remaining arguments is the same as for *GrabPointer*. The active grab is terminated automatically when the logical state of the pointer has all buttons released, independent of the logical state of modifier keys. Note that the logical state of a device (as seen by means of the protocol) may lag the physical state if device event processing is frozen. This request overrides all previous passive grabs by the same client on the same button/key combinations on the same window. A modifier of *AnyModifier* is equivalent to issuing the request for all possible modifier combinations (including the combination of no modifiers). It is not required that all specified modifiers have currently assigned keycodes. A button of *AnyButton* is equivalent to issuing the request for all possible buttons. Otherwise, it is not required that the button specified currently be assigned to a physical button.

An *Access* error is generated if some other client has already issued a *GrabButton* request with the same button/key combination on the same window. When using *AnyModifier* or *AnyButton*, the request fails completely (no grabs are established), and an *Access* error is generated if there is a conflicting grab for any combination. The request has no effect on an active grab.

RETURN VALUE

Returns an *xcb_void_cookie_t*. Errors (if any) have to be handled in the event loop.

If you want to handle errors directly with *xcb_request_check* instead, use *xcb_grab_button_checked*. See **xcb-requests(3)** for details.

ERRORS

xcb_access_error_t

Another client has already issued a GrabButton with the same button/key combination on the same window.

xcb_cursor_error_t

The specified *cursor* does not exist.

xcb_value_error_t

TODO: reasons?

xcb_window_error_t

The specified *window* does not exist.

SEE ALSO

xcb-requests(3)

AUTHOR

Generated from xproto.xml. Contact xcb@lists.freedesktop.org for corrections and improvements.