## NAME

xcb_grab_pointer - Grab the pointer

## SYNOPSIS

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

### Request function

xcb_grab_pointer_cookie_t **xcb_grab_pointer**(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_, xcb_timestamp_t _time_);

### Reply datastructure

```
typedef struct xcb_grab_pointer_reply_t {
   uint8_t  response_type;
   uint8_t  status;
   uint16_t sequence;
   uint32_t length;
} xcb_grab_pointer_reply_t;
```

### Reply function

xcb_grab_pointer_reply_t ***xcb_grab_pointer_reply**(xcb_connection_t *_conn_, xcb_grab_pointer_cookie_t _cookie_, xcb_generic_error_t ***_e_);

## 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.

*time* The time argument allows you to avoid certain circumstances that come up if applications take a long time to respond or if there are long network delays. Consider a situation where you have two applications, both of which normally grab the pointer when clicked on. If both applications specify the timestamp from the event, the second application may wake up faster and successfully grab the pointer before the first application. The first application then will get an indication that the other application grabbed the pointer before its request was processed.

The special value *XCB_CURRENT_TIME* will be replaced with the current server time.

**REPLY FIELDS**

*response_type* The type of this reply, in this case *XCB_GRAB_POINTER*. This field is also present in the *xcb_generic_reply_t* and can be used to tell replies apart from each other.

*sequence* The sequence number of the last request processed by the X11 server.

*length* The length of the reply, in words (a word is 4 bytes).

*status* One of the following values:

*XCB_GRAB_STATUS_SUCCESS*
TODO: NOT YET DOCUMENTED.

*XCB_GRAB_STATUS_ALREADY_GRABBED*
TODO: NOT YET DOCUMENTED.

*XCB_GRAB_STATUS_INVALID_TIME*
TODO: NOT YET DOCUMENTED.

*XCB_GRAB_STATUS_NOT_VIEWABLE*
TODO: NOT YET DOCUMENTED.

*XCB_GRAB_STATUS_FROZEN*
TODO: NOT YET DOCUMENTED.
TODO: NOT YET DOCUMENTED.

**DESCRIPTION**

Actively grabs control of the pointer. Further pointer events are reported only to the grabbing client. Overrides any active pointer grab by this client.

**RETURN VALUE**

Returns an *xcb_grab_pointer_cookie_t*. Errors have to be handled when calling the reply function *xcb_grab_pointer_reply*.

If you want to handle errors in the event loop instead, use *xcb_grab_pointer_unchecked*. See **xcb-requests(3)** for details.

**ERRORS**

*xcb_value_error_t*
TODO: reasons?

*xcb_window_error_t*
The specified *window* does not exist.

**EXAMPLE**

```
/*
 * Grabs the pointer actively
 *
 */
void my_example(xcb_connection_t *conn, xcb_screen_t *screen, xcb_cursor_t cursor) {
   xcb_grab_pointer_cookie_t cookie;
   xcb_grab_pointer_reply_t *reply;

   cookie = xcb_grab_pointer(
      conn,
      false,            /* get all pointer events specified by the following mask */
      screen->root,     /* grab the root window */
      XCB_NONE,         /* which events to let through */
      XCB_GRAB_MODE_ASYNC, /* pointer events should continue as normal */
      XCB_GRAB_MODE_ASYNC, /* keyboard mode */
      XCB_NONE,         /* confine_to = in which window should the cursor stay */
      cursor,           /* we change the cursor to whatever the user wanted */
      XCB_CURRENT_TIME
   );

   if ((reply = xcb_grab_pointer_reply(conn, cookie, NULL))) {
      if (reply->status == XCB_GRAB_STATUS_SUCCESS)
         printf("successfully grabbed the pointer\n");
      free(reply);
   }
}
```

**SEE ALSO**

**xcb-requests**(3), **xcb-examples**(3), **xcb_grab_keyboard**(3)

**AUTHOR**

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