

**NAME**

xcb\_input\_xi\_query\_pointer -

**SYNOPSIS**

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

**Request function**

```
xcb_input_xi_query_pointer_cookie_t xcb_input_xi_query_pointer(xcb_connection_t *conn,
    xcb_window_t window, xcb_input_device_id_t deviceid);
```

**Reply datastructure**

```
typedef struct xcb_input_xi_query_pointer_reply_t {
    uint8_t      response_type;
    uint8_t      pad0;
    uint16_t     sequence;
    uint32_t     length;
    xcb_window_t root;
    xcb_window_t child;
    xcb_input_fp1616_t root_x;
    xcb_input_fp1616_t root_y;
    xcb_input_fp1616_t win_x;
    xcb_input_fp1616_t win_y;
    uint8_t      same_screen;
    uint8_t      pad1;
    uint16_t     buttons_len;
    xcb_input_modifier_info_t mods;
    xcb_input_group_info_t group;
} xcb_input_xi_query_pointer_reply_t;
```

**Reply function**

```
xcb_input_xi_query_pointer_reply_t *xcb_input_xi_query_pointer_reply(xcb_connection_t *conn,
    xcb_input_xi_query_pointer_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
uint32_t *xcb_input_xi_query_pointer_buttons(const xcb_input_xi_query_pointer_request_t *reply);
```

```
int xcb_input_xi_query_pointer_buttons_length(const xcb_input_xi_query_pointer_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_input_xi_query_pointer_buttons_end(const
```

```
xcb_input_xi_query_pointer_reply_t *reply);
```

## REQUEST ARGUMENTS

|                 |                            |
|-----------------|----------------------------|
| <i>conn</i>     | The XCB connection to X11. |
| <i>window</i>   | TODO: NOT YET DOCUMENTED.  |
| <i>deviceid</i> | TODO: NOT YET DOCUMENTED.  |

## REPLY FIELDS

|                      |  |
|----------------------|--|
| <i>response_type</i> | The type of this reply, in this case <i>XCB_INPUT_XI_QUERY_POINTER</i> . This field is also present in the <i>xcb_generic_reply_t</i> and can be used to tell replies apart from each other. |
| <i>sequence</i>      | The sequence number of the last request processed by the X11 server.   |
| <i>length</i>        | The length of the reply, in words (a word is 4 bytes).   |
| <i>root</i>          | TODO: NOT YET DOCUMENTED.  |
| <i>child</i>         | TODO: NOT YET DOCUMENTED.  |
| <i>root_x</i>        | TODO: NOT YET DOCUMENTED.  |
| <i>root_y</i>        | TODO: NOT YET DOCUMENTED.  |
| <i>win_x</i>         | TODO: NOT YET DOCUMENTED.  |
| <i>win_y</i>         | TODO: NOT YET DOCUMENTED.  |
| <i>same_screen</i>   | TODO: NOT YET DOCUMENTED.  |
| <i>buttons_len</i>   | TODO: NOT YET DOCUMENTED.  |
| <i>mods</i>          | TODO: NOT YET DOCUMENTED.  |
| <i>group</i>         | TODO: NOT YET DOCUMENTED.  |

## DESCRIPTION

## RETURN VALUE

Returns an *xcb\_input\_xi\_query\_pointer\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_input\_xi\_query\_pointer\_reply*.

If you want to handle errors in the event loop instead, use *xcb\_input\_xi\_query\_pointer\_unchecked*. See **xcb-requests(3)** for details.

**ERRORS**

This request does never generate any errors.

**SEE ALSO****AUTHOR**

Generated from xinput.xml. Contact [xcb@lists.freedesktop.org](mailto:xcb@lists.freedesktop.org) for corrections and improvements.