

**NAME**

xcb\_test\_compare\_cursor -

**SYNOPSIS**

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

**Request function**

```
xcb_test_compare_cursor_cookie_t xcb_test_compare_cursor(xcb_connection_t *conn,  
xcb_window_t window, xcb_cursor_t cursor);
```

**Reply datastructure**

```
typedef struct xcb_test_compare_cursor_reply_t {  
    uint8_t response_type;  
    uint8_t same;  
    uint16_t sequence;  
    uint32_t length;  
} xcb_test_compare_cursor_reply_t;
```

**Reply function**

```
xcb_test_compare_cursor_reply_t *xcb_test_compare_cursor_reply(xcb_connection_t *conn,  
xcb_test_compare_cursor_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*window*         TODO: NOT YET DOCUMENTED.

*cursor*         TODO: NOT YET DOCUMENTED.

**REPLY FIELDS**

*response\_type*   The type of this reply, in this case *XCB\_TEST\_COMPARE\_CURSOR*. 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).

*same*            TODO: NOT YET DOCUMENTED.

**DESCRIPTION****RETURN VALUE**

Returns an *xcb\_test\_compare\_cursor\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_test\_compare\_cursor\_reply*.

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

**ERRORS**

This request does never generate any errors.

**SEE ALSO****AUTHOR**

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