

**NAME**

xcb\_get\_selection\_owner - Gets the owner of a selection

**SYNOPSIS**

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

**Request function**

```
xcb_get_selection_owner_cookie_t xcb_get_selection_owner(xcb_connection_t *conn,  
xcb_atom_t selection);
```

**Reply datastructure**

```
typedef struct xcb_get_selection_owner_reply_t {  
    uint8_t    response_type;  
    uint8_t    pad0;  
    uint16_t   sequence;  
    uint32_t   length;  
    xcb_window_t owner;  
} xcb_get_selection_owner_reply_t;
```

**Reply function**

```
xcb_get_selection_owner_reply_t *xcb_get_selection_owner_reply(xcb_connection_t *conn,  
xcb_get_selection_owner_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*selection*       The selection.

**REPLY FIELDS**

*response\_type*   The type of this reply, in this case *XCB\_GET\_SELECTION\_OWNER*. 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).

*owner*            The current selection owner window.

**DESCRIPTION**

Gets the owner of the specified selection.

TODO: briefly explain what a selection is.

**RETURN VALUE**

Returns an *xcb\_get\_selection\_owner\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_get\_selection\_owner\_reply*.

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

**ERRORS**

*xcb\_atom\_error\_t*

*selection* does not refer to a valid atom.

**SEE ALSO**

**xcb-requests(3)**, **xcb\_set\_selection\_owner(3)**

**AUTHOR**

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