

**NAME**

xcb\_get\_pointer\_control -

**SYNOPSIS**

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

**Request function**

```
xcb_get_pointer_control_cookie_t xcb_get_pointer_control(xcb_connection_t *conn,
```

**Reply datastructure**

```
typedef struct xcb_get_pointer_control_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint16_t acceleration_numerator;
    uint16_t acceleration_denominator;
    uint16_t threshold;
    uint8_t pad1[18];
} xcb_get_pointer_control_reply_t;
```

**Reply function**

```
xcb_get_pointer_control_reply_t *xcb_get_pointer_control_reply(xcb_connection_t *conn,
    xcb_get_pointer_control_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

*conn*            The XCB connection to X11.

**REPLY FIELDS**

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

*acceleration\_numerator*  
                  TODO: NOT YET DOCUMENTED.

*acceleration\_denominator*

TODO: NOT YET DOCUMENTED.

*threshold*

TODO: NOT YET DOCUMENTED.

## DESCRIPTION

## RETURN VALUE

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

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

## ERRORS

This request does never generate any errors.

## SEE ALSO

## AUTHOR

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