NAME

xcb_input_get_device_control -

SYNOPSIS

#include <xcb/xinput.h>

Request function

xcb_input_get_device_control_cookie_t **xcb_input_get_device_control**(xcb_connection_t **conn*, uint16_t *control_id*, uint8_t *device_id*);

Reply datastructure

```
typedef struct xcb_input_get_device_control_reply_t {
    uint8_t  response_type;
    uint8_t  xi_reply_type;
    uint16_t sequence;
    uint32_t length;
    uint8_t  status;
    uint8_t  pad0[23];
} xcb_input_get_device_control_reply_t;
```

Reply function

xcb_input_get_device_control_reply_t ***xcb_input_get_device_control_reply**(xcb_connection_t **conn*, xcb_input_get_device_control_cookie_t *cookie*, xcb_generic_error_t ***e*);

Reply accessors

xcb_input_device_state_t *xcb_input_get_device_control_control (const xcb_input_get_device_control_request_t *reply)

REQUEST ARGUMENTS

conn The XCB connection to X11.

control_id TODO: NOT YET DOCUMENTED.

device_id TODO: NOT YET DOCUMENTED.

REPLY FIELDS

response_type The type of this reply, in this case *XCB_INPUT_GET_DEVICE_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).

xi_reply_type TODO: NOT YET DOCUMENTED.

status TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_input_get_device_control_cookie_t*. Errors have to be handled when calling the reply function *xcb_input_get_device_control_reply*.

If you want to handle errors in the event loop instead, use *xcb_input_get_device_control_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 for corrections and improvements.