

NAME

xcb_input_get_device_property -

SYNOPSIS

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

Request function

```
xcb_input_get_device_property_cookie_t xcb_input_get_device_property(xcb_connection_t *conn,  
    xcb_atom_t property, xcb_atom_t type, uint32_t offset, uint32_t len, uint8_t device_id,  
    uint8_t _delete);
```

Reply datastructure

```
typedef struct xcb_input_get_device_property_reply_t {  
    uint8_t response_type;  
    uint8_t xi_reply_type;  
    uint16_t sequence;  
    uint32_t length;  
    xcb_atom_t type;  
    uint32_t bytes_after;  
    uint32_t num_items;  
    uint8_t format;  
    uint8_t device_id;  
    uint8_t pad0[10];  
} xcb_input_get_device_property_reply_t;
```

Reply function

```
xcb_input_get_device_property_reply_t  
    *xcb_input_get_device_property_reply(xcb_connection_t *conn,  
    xcb_input_get_device_property_cookie_t cookie, xcb_generic_error_t **e);
```

Reply accessors

```
xcb_input_get_device_property_items_t *xcb_input_get_device_property_items (const  
    xcb_input_get_device_property_request_t *reply)
```

REQUEST ARGUMENTS

conn The XCB connection to X11.

property TODO: NOT YET DOCUMENTED.

<i>type</i>	TODO: NOT YET DOCUMENTED.
<i>offset</i>	TODO: NOT YET DOCUMENTED.
<i>len</i>	TODO: NOT YET DOCUMENTED.
<i>device_id</i>	TODO: NOT YET DOCUMENTED.
<i>_delete</i>	TODO: NOT YET DOCUMENTED.

REPLY FIELDS

<i>response_type</i>	The type of this reply, in this case <i>XCB_INPUT_GET_DEVICE_PROPERTY</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>xi_reply_type</i>	TODO: NOT YET DOCUMENTED.
<i>type</i>	TODO: NOT YET DOCUMENTED.
<i>bytes_after</i>	TODO: NOT YET DOCUMENTED.
<i>num_items</i>	TODO: NOT YET DOCUMENTED.
<i>format</i>	TODO: NOT YET DOCUMENTED.
<i>device_id</i>	TODO: NOT YET DOCUMENTED.

DESCRIPTION**RETURN VALUE**

Returns an *xcb_input_get_device_property_cookie_t*. Errors have to be handled when calling the reply function *xcb_input_get_device_property_reply*.

If you want to handle errors in the event loop instead, use *xcb_input_get_device_property_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.