

NAME

xcb_glx_is_enabled -

SYNOPSIS

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

Request function

```
xcb_glx_is_enabled_cookie_t xcb_glx_is_enabled(xcb_connection_t *conn,
        xcb_glx_context_tag_t context_tag, uint32_t capability);
```

Reply datastructure

```
typedef struct xcb_glx_is_enabled_reply_t {
    uint8_t    response_type;
    uint8_t    pad0;
    uint16_t   sequence;
    uint32_t   length;
    xcb_glx_bool32_t ret_val;
} xcb_glx_is_enabled_reply_t;
```

Reply function

```
xcb_glx_is_enabled_reply_t *xcb_glx_is_enabled_reply(xcb_connection_t *conn,
        xcb_glx_is_enabled_cookie_t cookie, xcb_generic_error_t **e);
```

REQUEST ARGUMENTS

conn The XCB connection to X11.

context_tag TODO: NOT YET DOCUMENTED.

capability TODO: NOT YET DOCUMENTED.

REPLY FIELDS

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

ret_val TODO: NOT YET DOCUMENTED.

DESCRIPTION**RETURN VALUE**

Returns an *xcb_glx_is_enabled_cookie_t*. Errors have to be handled when calling the reply function *xcb_glx_is_enabled_reply*.

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

ERRORS

This request does never generate any errors.

SEE ALSO**AUTHOR**

Generated from glx.xml. Contact xcb@lists.freedesktop.org for corrections and improvements.