

**NAME**

xcb\_glx\_is\_texture -

**SYNOPSIS**

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

**Request function**

```
xcb_glx_is_texture_cookie_t xcb_glx_is_texture(xcb_connection_t *conn,
        xcb_glx_context_tag_t context_tag, uint32_t texture);
```

**Reply datastructure**

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

**Reply function**

```
xcb_glx_is_texture_reply_t *xcb_glx_is_texture_reply(xcb_connection_t *conn,
        xcb_glx_is_texture_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*context\_tag*    TODO: NOT YET DOCUMENTED.

*texture*        TODO: NOT YET DOCUMENTED.

**REPLY FIELDS**

*response\_type* The type of this reply, in this case *XCB\_GLX\_IS\_TEXTURE*. 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\_texture\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_glx\_is\_texture\_reply*.

If you want to handle errors in the event loop instead, use *xcb\_glx\_is\_texture\_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](mailto:xcb@lists.freedesktop.org) for corrections and improvements.