

**NAME**

xcb\_glx\_read\_pixels -

**SYNOPSIS**

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

**Request function**

```
xcb_glx_read_pixels_cookie_t xcb_glx_read_pixels(xcb_connection_t *conn,
        xcb_glx_context_tag_t context_tag, int32_t x, int32_t y, int32_t width, int32_t height,
        uint32_t format, uint32_t type, uint8_t swap_bytes, uint8_t lsb_first);
```

**Reply datastructure**

```
typedef struct xcb_glx_read_pixels_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint8_t pad1[24];
} xcb_glx_read_pixels_reply_t;
```

**Reply function**

```
xcb_glx_read_pixels_reply_t *xcb_glx_read_pixels_reply(xcb_connection_t *conn,
        xcb_glx_read_pixels_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
uint8_t *xcb_glx_read_pixels_data(const xcb_glx_read_pixels_request_t *reply);
```

```
int xcb_glx_read_pixels_data_length(const xcb_glx_read_pixels_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_glx_read_pixels_data_end(const xcb_glx_read_pixels_reply_t *reply);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*context\_tag*    TODO: NOT YET DOCUMENTED.

*x*                TODO: NOT YET DOCUMENTED.

*y*                TODO: NOT YET DOCUMENTED.

<i>width</i>	TODO: NOT YET DOCUMENTED.
<i>height</i>	TODO: NOT YET DOCUMENTED.
<i>format</i>	TODO: NOT YET DOCUMENTED.
<i>type</i>	TODO: NOT YET DOCUMENTED.
<i>swap_bytes</i>	TODO: NOT YET DOCUMENTED.
<i>lsb_first</i>	TODO: NOT YET DOCUMENTED.

## REPLY FIELDS

<i>response_type</i>	The type of this reply, in this case <i>XCB_GLX_READ_PIXELS</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).

## DESCRIPTION

### RETURN VALUE

Returns an *xcb\_glx\_read\_pixels\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_glx\_read\_pixels\_reply*.

If you want to handle errors in the event loop instead, use *xcb\_glx\_read\_pixels\_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.