

**NAME**

xcb\_glx\_vendor\_private\_with\_reply -

**SYNOPSIS**

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

**Request function**

```
xcb_glx_vendor_private_with_reply_cookie_t
xcb_glx_vendor_private_with_reply(xcb_connection_t *conn, uint32_t vendor_code,
xcb_glx_context_tag_t context_tag, uint32_t data_len, const uint8_t *data);
```

**Reply datastructure**

```
typedef struct xcb_glx_vendor_private_with_reply_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint32_t retval;
    uint8_t data1[24];
} xcb_glx_vendor_private_with_reply_reply_t;
```

**Reply function**

```
xcb_glx_vendor_private_with_reply_reply_t
*xcb_glx_vendor_private_with_reply_reply(xcb_connection_t *conn,
xcb_glx_vendor_private_with_reply_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
uint8_t *xcb_glx_vendor_private_with_reply_data_2(const
xcb_glx_vendor_private_with_reply_request_t *reply);

int xcb_glx_vendor_private_with_reply_data_2_length(const
xcb_glx_vendor_private_with_reply_reply_t *reply);

xcb_generic_iterator_t xcb_glx_vendor_private_with_reply_data_2_end(const
xcb_glx_vendor_private_with_reply_reply_t *reply);
```

**REQUEST ARGUMENTS**

*conn*            The XCB connection to X11.

xcb\_glx\_vendor\_private\_with\_reply(3)      XCB Requests      xcb\_glx\_vendor\_private\_with\_reply(3)

*vendor\_code*      TODO: NOT YET DOCUMENTED.

*context\_tag*      TODO: NOT YET DOCUMENTED.

*data\_len*      TODO: NOT YET DOCUMENTED.

*data*      TODO: NOT YET DOCUMENTED.

## REPLY FIELDS

*response\_type*      The type of this reply, in this case *XCB\_GLX\_VENDOR\_PRIVATE\_WITH\_REPLY*. 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).

*retval*      TODO: NOT YET DOCUMENTED.

*data1*      TODO: NOT YET DOCUMENTED.

## DESCRIPTION

### RETURN VALUE

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

If you want to handle errors in the event loop instead, use *xcb\_glx\_vendor\_private\_with\_reply\_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.