

NAME

xcb_xf86dri_get_client_driver_name -

SYNOPSIS**#include** <xcb/xf86dri.h>**Request function**

```
xcb_xf86dri_get_client_driver_name_cookie_t
xcb_xf86dri_get_client_driver_name(xcb_connection_t *conn, uint32_t screen);
```

Reply datastructure

```
typedef struct xcb_xf86dri_get_client_driver_name_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint32_t client_driver_major_version;
    uint32_t client_driver_minor_version;
    uint32_t client_driver_patch_version;
    uint32_t client_driver_name_len;
    uint8_t pad1[8];
} xcb_xf86dri_get_client_driver_name_reply_t;
```

Reply function

```
xcb_xf86dri_get_client_driver_name_reply_t
*xcb_xf86dri_get_client_driver_name_reply(xcb_connection_t *conn,
xcb_xf86dri_get_client_driver_name_cookie_t cookie, xcb_generic_error_t **e);
```

Reply accessors

```
char *xcb_xf86dri_get_client_driver_name_client_driver_name(const
xcb_xf86dri_get_client_driver_name_request_t *reply);

int xcb_xf86dri_get_client_driver_name_client_driver_name_length(const
xcb_xf86dri_get_client_driver_name_reply_t *reply);

xcb_generic_iterator_t xcb_xf86dri_get_client_driver_name_client_driver_name_end(const
xcb_xf86dri_get_client_driver_name_reply_t *reply);
```

REQUEST ARGUMENTS

xcb_xf86dri_get_client_driver_name(3) XCB Requests xcb_xf86dri_get_client_driver_name(3)

conn The XCB connection to X11.

screen TODO: NOT YET DOCUMENTED.

REPLY FIELDS

response_type The type of this reply, in this case *XCB_XF86DRI_GET_CLIENT_DRIVER_NAME*. 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).

client_driver_major_version
TODO: NOT YET DOCUMENTED.

client_driver_minor_version
TODO: NOT YET DOCUMENTED.

client_driver_patch_version
TODO: NOT YET DOCUMENTED.

client_driver_name_len
TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_xf86dri_get_client_driver_name_cookie_t*. Errors have to be handled when calling the reply function *xcb_xf86dri_get_client_driver_name_reply*.

If you want to handle errors in the event loop instead, use *xcb_xf86dri_get_client_driver_name_unchecked*. See **xcb-requests(3)** for details.

ERRORS

This request does never generate any errors.

SEE ALSO

AUTHOR

Generated from xf86dri.xml. Contact xcb@lists.freedesktop.org for corrections and improvements.