

**NAME**

xcb\_xvnc\_create\_context -

**SYNOPSIS**

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

**Request function**

```
xcb_xvnc_create_context_cookie_t xcb_xvnc_create_context(xcb_connection_t *conn,
    xcb_xvnc_context_t context_id, xcb_xv_port_t port_id, xcb_xvnc_surface_t surface_id,
    uint16_t width, uint16_t height, uint32_t flags);
```

**Reply datastructure**

```
typedef struct xcb_xvnc_create_context_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint16_t width_actual;
    uint16_t height_actual;
    uint32_t flags_return;
    uint8_t pad1[20];
} xcb_xvnc_create_context_reply_t;
```

**Reply function**

```
xcb_xvnc_create_context_reply_t *xcb_xvnc_create_context_reply(xcb_connection_t *conn,
    xcb_xvnc_create_context_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
uint32_t *xcb_xvnc_create_context_priv_data(const xcb_xvnc_create_context_request_t *reply);
```

```
int xcb_xvnc_create_context_priv_data_length(const xcb_xvnc_create_context_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_xvnc_create_context_priv_data_end(const
    xcb_xvnc_create_context_reply_t *reply);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*context\_id*     TODO: NOT YET DOCUMENTED.

<i>port_id</i>	TODO: NOT YET DOCUMENTED.
<i>surface_id</i>	TODO: NOT YET DOCUMENTED.
<i>width</i>	TODO: NOT YET DOCUMENTED.
<i>height</i>	TODO: NOT YET DOCUMENTED.
<i>flags</i>	TODO: NOT YET DOCUMENTED.

## REPLY FIELDS

<i>response_type</i>	The type of this reply, in this case <i>XCB_XVNC_CREATE_CONTEXT</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).
<i>width_actual</i>	TODO: NOT YET DOCUMENTED.
<i>height_actual</i>	TODO: NOT YET DOCUMENTED.
<i>flags_return</i>	TODO: NOT YET DOCUMENTED.

## DESCRIPTION

### RETURN VALUE

Returns an *xcb\_xvnc\_create\_context\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_xvnc\_create\_context\_reply*.

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

## ERRORS

This request does never generate any errors.

## SEE ALSO

### AUTHOR

Generated from xvnc.xml. Contact [xcb@lists.freedesktop.org](mailto:xcb@lists.freedesktop.org) for corrections and improvements.