

**NAME**

xcb\_render\_query\_pict\_formats -

**SYNOPSIS**

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

**Request function**

```
xcb_render_query_pict_formats_cookie_t xcb_render_query_pict_formats(xcb_connection_t *conn,
```

**Reply datastructure**

```
typedef struct xcb_render_query_pict_formats_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint32_t num_formats;
    uint32_t num_screens;
    uint32_t num_depths;
    uint32_t num_visuals;
    uint32_t num_subpixel;
    uint8_t pad1[4];
} xcb_render_query_pict_formats_reply_t;
```

**Reply function**

```
xcb_render_query_pict_formats_reply_t
*xcb_render_query_pict_formats_reply(xcb_connection_t *conn,
    xcb_render_query_pict_formats_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
xcb_render_pictforminfo_t *xcb_render_query_pict_formats_formats(const
    xcb_render_query_pict_formats_request_t *reply);
```

```
int xcb_render_query_pict_formats_formats_length(const xcb_render_query_pict_formats_reply_t
    *reply);
```

```
xcb_render_pictforminfo_iterator_t xcb_render_query_pict_formats_formats_iterator(const
    xcb_render_query_pict_formats_reply_t *reply);
```

```
int xcb_render_query_pict_formats_screens_length(const xcb_render_query_pict_formats_reply_t
```

```
*reply);
```

```
xcb_render_pictscreen_iterator_t xcb_render_query_pict_formats_screens_iterator(const
    xcb_render_query_pict_formats_reply_t *reply);
```

```
uint32_t *xcb_render_query_pict_formats_subpixels(const xcb_render_query_pict_formats_request_t
    *reply);
```

```
int xcb_render_query_pict_formats_subpixels_length(const xcb_render_query_pict_formats_reply_t
    *reply);
```

```
xcb_generic_iterator_t xcb_render_query_pict_formats_subpixels_end(const
    xcb_render_query_pict_formats_reply_t *reply);
```

## REQUEST ARGUMENTS

*conn*            The XCB connection to X11.

## REPLY FIELDS

*response\_type*    The type of this reply, in this case *XCB\_RENDER\_QUERY\_PICT\_FORMATS*. 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).

*num\_formats*      TODO: NOT YET DOCUMENTED.

*num\_screens*      TODO: NOT YET DOCUMENTED.

*num\_depths*        TODO: NOT YET DOCUMENTED.

*num\_visuals*        TODO: NOT YET DOCUMENTED.

*num\_subpixel*      TODO: NOT YET DOCUMENTED.

## DESCRIPTION

### RETURN VALUE

Returns an *xcb\_render\_query\_pict\_formats\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_render\_query\_pict\_formats\_reply*.

If you want to handle errors in the event loop instead, use *xcb\_render\_query\_pict\_formats\_unchecked*. See **xcb-requests(3)** for details.

**ERRORS**

This request does never generate any errors.

**SEE ALSO****AUTHOR**

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