

**NAME**

`xcb_query_extension` - check if extension is present

**SYNOPSIS**

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

**Request function**

```
xcb_query_extension_cookie_t xcb_query_extension(xcb_connection_t *conn, uint16_t name_len,
const char *name);
```

**Reply datastructure**

```
typedef struct xcb_query_extension_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint8_t present;
    uint8_t major_opcode;
    uint8_t first_event;
    uint8_t first_error;
} xcb_query_extension_reply_t;
```

**Reply function**

```
xcb_query_extension_reply_t *xcb_query_extension_reply(xcb_connection_t *conn,
xcb_query_extension_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*name\_len*       The length of *name* in bytes.

*name*           The name of the extension to query, for example "RANDR". This is case sensitive!

**REPLY FIELDS**

*response\_type*   The type of this reply, in this case `XCB_QUERY_EXTENSION`. 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.

<i>length</i>	The length of the reply, in words (a word is 4 bytes).
<i>present</i>	Whether the extension is present on this X11 server.
<i>major_opcode</i>	The major opcode for requests.
<i>first_event</i>	The first event code, if any.
<i>first_error</i>	The first error code, if any.

## DESCRIPTION

Determines if the specified extension is present on this X11 server.

Every extension has a unique *major\_opcode* to identify requests, the minor opcodes and request formats are extension-specific. If the extension provides events and errors, the *first\_event* and *first\_error* fields in the reply are set accordingly.

There should rarely be a need to use this request directly, XCB provides the *xcb\_get\_extension\_data* function instead.

## RETURN VALUE

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

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

## ERRORS

This request does never generate any errors.

## SEE ALSO

**xcb-requests(3)**, **xcb\_get\_extension\_data(3)**, **xdpyinfo(1)**

## AUTHOR

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