

**NAME**

xcb\_sync\_query\_counter -

**SYNOPSIS**

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

**Request function**

```
xcb_sync_query_counter_cookie_t xcb_sync_query_counter(xcb_connection_t *conn,
    xcb_sync_counter_t counter);
```

**Reply datastructure**

```
typedef struct xcb_sync_query_counter_reply_t {
    uint8_t      response_type;
    uint8_t      pad0;
    uint16_t     sequence;
    uint32_t     length;
    xcb_sync_int64_t counter_value;
} xcb_sync_query_counter_reply_t;
```

**Reply function**

```
xcb_sync_query_counter_reply_t *xcb_sync_query_counter_reply(xcb_connection_t *conn,
    xcb_sync_query_counter_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

*conn*            The XCB connection to X11.

*counter*        TODO: NOT YET DOCUMENTED.

**REPLY FIELDS**

*response\_type*   The type of this reply, in this case *XCB\_SYNC\_QUERY\_COUNTER*. 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).

*counter\_value*   TODO: NOT YET DOCUMENTED.

**DESCRIPTION****RETURN VALUE**

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

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

**ERRORS**

This request does never generate any errors.

**SEE ALSO****AUTHOR**

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