

**NAME**

xcb\_sync\_query\_fence -

**SYNOPSIS**

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

**Request function**

```
xcb_sync_query_fence_cookie_t xcb_sync_query_fence(xcb_connection_t *conn,  
xcb_sync_fence_t fence);
```

**Reply datastructure**

```
typedef struct xcb_sync_query_fence_reply_t {  
    uint8_t response_type;  
    uint8_t pad0;  
    uint16_t sequence;  
    uint32_t length;  
    uint8_t triggered;  
    uint8_t pad1[23];  
} xcb_sync_query_fence_reply_t;
```

**Reply function**

```
xcb_sync_query_fence_reply_t *xcb_sync_query_fence_reply(xcb_connection_t *conn,  
xcb_sync_query_fence_cookie_t cookie, xcb_generic_error_t **e);
```

**REQUEST ARGUMENTS**

<i>conn</i>	The XCB connection to X11.
<i>fence</i>	TODO: NOT YET DOCUMENTED.

**REPLY FIELDS**

<i>response_type</i>	The type of this reply, in this case <i>XCB_SYNC_QUERY_FENCE</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>triggered</i>	TODO: NOT YET DOCUMENTED.

**DESCRIPTION****RETURN VALUE**

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

If you want to handle errors in the event loop instead, use *xcb\_sync\_query\_fence\_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.