

**NAME**

xcb\_query\_tree - query the window tree

**SYNOPSIS**

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

**Request function**

```
xcb_query_tree_cookie_t xcb_query_tree(xcb_connection_t *conn, xcb_window_t window);
```

**Reply datastructure**

```
typedef struct xcb_query_tree_reply_t {
    uint8_t    response_type;
    uint8_t    pad0;
    uint16_t   sequence;
    uint32_t   length;
    xcb_window_t root;
    xcb_window_t parent;
    uint16_t   children_len;
    uint8_t    pad1[14];
} xcb_query_tree_reply_t;
```

**Reply function**

```
xcb_query_tree_reply_t *xcb_query_tree_reply(xcb_connection_t *conn,
    xcb_query_tree_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
xcb_window_t *xcb_query_tree_children(const xcb_query_tree_request_t *reply);
```

```
int xcb_query_tree_children_length(const xcb_query_tree_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_query_tree_children_end(const xcb_query_tree_reply_t *reply);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*window*         The *window* to query.

**REPLY FIELDS**

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

*root* The root window of *window*.

*parent* The parent window of *window*.

*children\_len* The number of child windows.

## DESCRIPTION

Gets the root window ID, parent window ID and list of children windows for the specified *window*. The children are listed in bottom-to-top stacking order.

## RETURN VALUE

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

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

## ERRORS

This request does never generate any errors.

## EXAMPLE

```
/*
 * Displays the root, parent and children of the specified window.
 *
 */
void my_example(xcb_connection_t *conn, xcb_window_t window) {
    xcb_query_tree_cookie_t cookie;
    xcb_query_tree_reply_t *reply;

    cookie = xcb_query_tree(conn, window);
    if ((reply = xcb_query_tree_reply(conn, cookie, NULL))) {
        printf("root = 0x%08x\n", reply->root);
        printf("parent = 0x%08x\n", reply->parent);
    }
}
```

```
xcb_window_t *children = xcb_query_tree_children(reply);
for (int i = 0; i < xcb_query_tree_children_length(reply); i++)
    printf("child window = 0x%08x\n", children[i]);

    free(reply);
}
}
```

**SEE ALSO**

**xcb-requests(3)**, **xcb-examples(3)**, **xwininfo(1)**

**AUTHOR**

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