

xcb\_xfixes\_get\_cursor\_image\_and\_name(3) XCB Requests xcb\_xfixes\_get\_cursor\_image\_and\_name(3)

## NAME

xcb\_xfixes\_get\_cursor\_image\_and\_name -

## SYNOPSIS

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

### Request function

```
xcb_xfixes_get_cursor_image_and_name_cookie_t  
xcb_xfixes_get_cursor_image_and_name(xcb_connection_t *conn,
```

### Reply datastructure

```
typedef struct xcb_xfixes_get_cursor_image_and_name_reply_t {  
    uint8_t response_type;  
    uint8_t pad0;  
    uint16_t sequence;  
    uint32_t length;  
    int16_t x;  
    int16_t y;  
    uint16_t width;  
    uint16_t height;  
    uint16_t xhot;  
    uint16_t yhot;  
    uint32_t cursor_serial;  
    xcb_atom_t cursor_atom;  
    uint16_t nbytes;  
    uint8_t pad1[2];  
} xcb_xfixes_get_cursor_image_and_name_reply_t;
```

### Reply function

```
xcb_xfixes_get_cursor_image_and_name_reply_t  
*xcb_xfixes_get_cursor_image_and_name_reply(xcb_connection_t *conn,  
xcb_xfixes_get_cursor_image_and_name_cookie_t cookie, xcb_generic_error_t **e);
```

### Reply accessors

```
uint32_t *xcb_xfixes_get_cursor_image_and_name_cursor_image(const  
xcb_xfixes_get_cursor_image_and_name_request_t *reply);  
  
int xcb_xfixes_get_cursor_image_and_name_cursor_image_length(const  
xcb_xfixes_get_cursor_image_and_name_reply_t *reply);
```

xcb\_xfixes\_get\_cursor\_image\_and\_name(3) XCB Requests xcb\_xfixes\_get\_cursor\_image\_and\_name(3)

```
xcb_generic_iterator_t xcb_xfixes_get_cursor_image_and_name_cursor_image_end(const
    xcb_xfixes_get_cursor_image_and_name_reply_t *reply);
```

```
char *xcb_xfixes_get_cursor_image_and_name_name(const
    xcb_xfixes_get_cursor_image_and_name_request_t *reply);
```

```
int xcb_xfixes_get_cursor_image_and_name_name_length(const
    xcb_xfixes_get_cursor_image_and_name_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_xfixes_get_cursor_image_and_name_name_end(const
    xcb_xfixes_get_cursor_image_and_name_reply_t *reply);
```

## REQUEST ARGUMENTS

*conn*            The XCB connection to X11.

## REPLY FIELDS

*response\_type*    The type of this reply, in this case  
*XCB\_XFIXES\_GET\_CURSOR\_IMAGE\_AND\_NAME*. 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).

*x*                TODO: NOT YET DOCUMENTED.

*y*                TODO: NOT YET DOCUMENTED.

*width*          TODO: NOT YET DOCUMENTED.

*height*         TODO: NOT YET DOCUMENTED.

*xhot*            TODO: NOT YET DOCUMENTED.

*yhot*            TODO: NOT YET DOCUMENTED.

*cursor\_serial*    TODO: NOT YET DOCUMENTED.

*cursor\_atom*    TODO: NOT YET DOCUMENTED.

xcb\_xfixes\_get\_cursor\_image\_and\_name(3) XCB Requests xcb\_xfixes\_get\_cursor\_image\_and\_name(3)

*nbytes*            TODO: NOT YET DOCUMENTED.

## **DESCRIPTION**

### **RETURN VALUE**

Returns an *xcb\_xfixes\_get\_cursor\_image\_and\_name\_cookie\_t*. Errors have to be handled when calling the reply function *xcb\_xfixes\_get\_cursor\_image\_and\_name\_reply*.

If you want to handle errors in the event loop instead, use *xcb\_xfixes\_get\_cursor\_image\_and\_name\_unchecked*. See **xcb-requests(3)** for details.

### **ERRORS**

This request does never generate any errors.

### **SEE ALSO**

### **AUTHOR**

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