NAME

xcb_randr_get_output_info -

SYNOPSIS

#include <xcb/randr.h>

Request function

xcb_randr_get_output_info_cookie_t **xcb_randr_get_output_info**(xcb_connection_t *_conn_, xcb_randr_output_t _output_, xcb_timestamp_t _config_timestamp_);

Reply datastructure

```
typedef struct xcb_randr_get_output_info_reply_t {
    uint8_t          response_type;
    uint8_t          status;
    uint16_t         sequence;
    uint32_t         length;
    xcb_timestamp_t  timestamp;
    xcb_randr_crtc_t crtc;
    uint32_t         mm_width;
    uint32_t         mm_height;
    uint8_t          connection;
    uint8_t          subpixel_order;
    uint16_t         num_crtcs;
    uint16_t         num_modes;
    uint16_t         num_preferred;
    uint16_t         num_clones;
    uint16_t         name_len;
} xcb_randr_get_output_info_reply_t;
```

Reply function

xcb_randr_get_output_info_reply_t ***xcb_randr_get_output_info_reply**(xcb_connection_t *_conn_, xcb_randr_get_output_info_cookie_t _cookie_, xcb_generic_error_t ***_e_);

Reply accessors

xcb_randr_crtc_t ***xcb_randr_get_output_info_crtcs**(const xcb_randr_get_output_info_request_t *_reply_);

int **xcb_randr_get_output_info_crtcs_length**(const xcb_randr_get_output_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_output_info_crtcs_end**(const xcb_randr_get_output_info_reply_t *_reply_);

xcb_randr_mode_t *__xcb_randr_get_output_info_modes__(const xcb_randr_get_output_info_request_t *_reply_);

int **xcb_randr_get_output_info_modes_length**(const xcb_randr_get_output_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_output_info_modes_end**(const xcb_randr_get_output_info_reply_t *_reply_);

xcb_randr_output_t *__xcb_randr_get_output_info_clones__(const xcb_randr_get_output_info_request_t *_reply_);

int **xcb_randr_get_output_info_clones_length**(const xcb_randr_get_output_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_output_info_clones_end**(const xcb_randr_get_output_info_reply_t *_reply_);

uint8_t *__xcb_randr_get_output_info_name__(const xcb_randr_get_output_info_request_t *_reply_);

int **xcb_randr_get_output_info_name_length**(const xcb_randr_get_output_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_output_info_name_end**(const xcb_randr_get_output_info_reply_t *_reply_);

REQUEST ARGUMENTS

conn The XCB connection to X11.

output TODO: NOT YET DOCUMENTED.

config_timestamp
TODO: NOT YET DOCUMENTED.

REPLY FIELDS

response_type The type of this reply, in this case *XCB_RANDR_GET_OUTPUT_INFO*. 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).

status TODO: NOT YET DOCUMENTED.

timestamp TODO: NOT YET DOCUMENTED.

crtc TODO: NOT YET DOCUMENTED.

mm_width TODO: NOT YET DOCUMENTED.

mm_height TODO: NOT YET DOCUMENTED.

connection TODO: NOT YET DOCUMENTED.

subpixel_order TODO: NOT YET DOCUMENTED.

num_crtcs TODO: NOT YET DOCUMENTED.

num_modes TODO: NOT YET DOCUMENTED.

num_preferred TODO: NOT YET DOCUMENTED.

num_clones TODO: NOT YET DOCUMENTED.

name_len TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_randr_get_output_info_cookie_t*. Errors have to be handled when calling the reply function *xcb_randr_get_output_info_reply*.

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

ERRORS

This request does never generate any errors.

SEE ALSO

AUTHOR

Generated from randr.xml. Contact xcb@lists.freedesktop.org for corrections and improvements.