NAME

xcb_randr_get_provider_info -

SYNOPSIS

#include <xcb/randr.h>

Request function

xcb_randr_get_provider_info_cookie_t **xcb_randr_get_provider_info**(xcb_connection_t *_conn_, xcb_randr_provider_t _provider_, xcb_timestamp_t _config_timestamp_);

Reply datastructure

```
typedef struct xcb_randr_get_provider_info_reply_t {
    uint8_t        response_type;
    uint8_t        status;
    uint16_t       sequence;
    uint32_t       length;
    xcb_timestamp_t timestamp;
    uint32_t       capabilities;
    uint16_t       num_crtcs;
    uint16_t       num_outputs;
    uint16_t       num_associated_providers;
    uint16_t       name_len;
    uint8_t        pad0[8];
} xcb_randr_get_provider_info_reply_t;
```

Reply function

xcb_randr_get_provider_info_reply_t ***xcb_randr_get_provider_info_reply**(xcb_connection_t *_conn_, xcb_randr_get_provider_info_cookie_t _cookie_, xcb_generic_error_t ***_e_);

Reply accessors

xcb_randr_crtc_t ***xcb_randr_get_provider_info_crtcs**(const xcb_randr_get_provider_info_request_t *_reply_);

int **xcb_randr_get_provider_info_crtcs_length**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_provider_info_crtcs_end**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_randr_output_t ***xcb_randr_get_provider_info_outputs**(const

xcb_randr_get_provider_info_request_t *_reply_);

int **xcb_randr_get_provider_info_outputs_length**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_provider_info_outputs_end**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_randr_provider_t ***xcb_randr_get_provider_info_associated_providers**(const xcb_randr_get_provider_info_request_t *_reply_);

int **xcb_randr_get_provider_info_associated_providers_length**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_provider_info_associated_providers_end**(const xcb_randr_get_provider_info_reply_t *_reply_);

uint32_t ***xcb_randr_get_provider_info_associated_capability**(const xcb_randr_get_provider_info_request_t *_reply_);

int **xcb_randr_get_provider_info_associated_capability_length**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_provider_info_associated_capability_end**(const xcb_randr_get_provider_info_reply_t *_reply_);

char ***xcb_randr_get_provider_info_name**(const xcb_randr_get_provider_info_request_t *_reply_);

int **xcb_randr_get_provider_info_name_length**(const xcb_randr_get_provider_info_reply_t *_reply_);

xcb_generic_iterator_t **xcb_randr_get_provider_info_name_end**(const xcb_randr_get_provider_info_reply_t *_reply_);

REQUEST ARGUMENTS

conn The XCB connection to X11.

provider 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_PROVIDER_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.

capabilities TODO: NOT YET DOCUMENTED.

num_crtcs TODO: NOT YET DOCUMENTED.

num_outputs TODO: NOT YET DOCUMENTED.

num_associated_providers
TODO: NOT YET DOCUMENTED.

name_len TODO: NOT YET DOCUMENTED.

DESCRIPTION

RETURN VALUE

Returns an *xcb_randr_get_provider_info_cookie_t*. Errors have to be handled when calling the reply function *xcb_randr_get_provider_info_reply*.

If you want to handle errors in the event loop instead, use *xcb_randr_get_provider_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.