

**NAME**

xcb\_randr\_query\_provider\_property -

**SYNOPSIS**

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

**Request function**

```
xcb_randr_query_provider_property_cookie_t  
xcb_randr_query_provider_property(xcb_connection_t *conn, xcb_randr_provider_t provider,  
xcb_atom_t property);
```

**Reply datastructure**

```
typedef struct xcb_randr_query_provider_property_reply_t {  
    uint8_t response_type;  
    uint8_t pad0;  
    uint16_t sequence;  
    uint32_t length;  
    uint8_t pending;  
    uint8_t range;  
    uint8_t immutable;  
    uint8_t pad1[21];  
} xcb_randr_query_provider_property_reply_t;
```

**Reply function**

```
xcb_randr_query_provider_property_reply_t  
*xcb_randr_query_provider_property_reply(xcb_connection_t *conn,  
xcb_randr_query_provider_property_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
int32_t *xcb_randr_query_provider_property_valid_values(const  
xcb_randr_query_provider_property_request_t *reply);
```

```
int xcb_randr_query_provider_property_valid_values_length(const  
xcb_randr_query_provider_property_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_randr_query_provider_property_valid_values_end(const  
xcb_randr_query_provider_property_reply_t *reply);
```

**REQUEST ARGUMENTS**

xcb\_randr\_query\_provider\_property(3)      XCB Requests      xcb\_randr\_query\_provider\_property(3)

*conn*      The XCB connection to X11.

*provider*      TODO: NOT YET DOCUMENTED.

*property*      TODO: NOT YET DOCUMENTED.

## REPLY FIELDS

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

*pending*      TODO: NOT YET DOCUMENTED.

*range*      TODO: NOT YET DOCUMENTED.

*immutable*      TODO: NOT YET DOCUMENTED.

## DESCRIPTION

### RETURN VALUE

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

If you want to handle errors in the event loop instead, use *xcb\_randr\_query\_provider\_property\_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](mailto:xcb@lists.freedesktop.org) for corrections and improvements.