

**NAME**

xcb\_list\_properties -

**SYNOPSIS**

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

**Request function**

```
xcb_list_properties_cookie_t xcb_list_properties(xcb_connection_t *conn, xcb_window_t window);
```

**Reply datastructure**

```
typedef struct xcb_list_properties_reply_t {
    uint8_t response_type;
    uint8_t pad0;
    uint16_t sequence;
    uint32_t length;
    uint16_t atoms_len;
    uint8_t pad1[22];
} xcb_list_properties_reply_t;
```

**Reply function**

```
xcb_list_properties_reply_t *xcb_list_properties_reply(xcb_connection_t *conn,
    xcb_list_properties_cookie_t cookie, xcb_generic_error_t **e);
```

**Reply accessors**

```
xcb_atom_t *xcb_list_properties_atoms(const xcb_list_properties_request_t *reply);
```

```
int xcb_list_properties_atoms_length(const xcb_list_properties_reply_t *reply);
```

```
xcb_generic_iterator_t xcb_list_properties_atoms_end(const xcb_list_properties_reply_t *reply);
```

**REQUEST ARGUMENTS**

*conn*           The XCB connection to X11.

*window*          TODO: NOT YET DOCUMENTED.

**REPLY FIELDS**

*response\_type*   The type of this reply, in this case *XCB\_LIST\_PROPERTIES*. This field is also present in the *xcb\_generic\_reply\_t* and can be used to tell replies apart from each other.

<i>sequence</i>	The sequence number of the last request processed by the X11 server.
<i>length</i>	The length of the reply, in words (a word is 4 bytes).
<i>atoms_len</i>	TODO: NOT YET DOCUMENTED.

**DESCRIPTION****RETURN VALUE**

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

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

**ERRORS**

This request does never generate any errors.

**SEE ALSO****AUTHOR**

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