NAME

xcb_send_event - send an event

SYNOPSIS

#include <xcb/xproto.h>

Request function

xcb_void_cookie_t **xcb_send_event**(xcb_connection_t *_conn_, uint8_t _propagate_,
xcb_window_t _destination_, uint32_t _event_mask_, const char *_event_);

REQUEST ARGUMENTS

conn The XCB connection to X11.

propagate If _propagate_ is true and no clients have selected any event on _destination_, the destination is replaced with the closest ancestor of _destination_ for which some client has selected a type in _event_mask_ and for which no intervening window has that type in its do-not-propagate-mask. If no such window exists or if the window is an ancestor of the focus window and _InputFocus_ was originally specified as the destination, the event is not sent to any clients. Otherwise, the event is reported to every client selecting on the final destination any of the types specified in _event_mask_.

destination The window to send this event to. Every client which selects any event within _event_mask_ on _destination_ will get the event.

The special value _XCB_SEND_EVENT_DEST_POINTER_WINDOW_ refers to the window that contains the mouse pointer.

The special value _XCB_SEND_EVENT_DEST_ITEM_FOCUS_ refers to the window which has the keyboard focus.

_event_mask_ Event_mask for determining which clients should receive the specified event. See _destination_ and _propagate_.

event The event to send to the specified _destination_.

DESCRIPTION

Identifies the _destination_ window, determines which clients should receive the specified event and ignores any active grabs.

The _event_ must be one of the core events or an event defined by an extension, so that the X server can

correctly byte-swap the contents as necessary. The contents of *event* are otherwise unaltered and unchecked except for the *send_event* field which is forced to 'true'.

RETURN VALUE

Returns an *xcb_void_cookie_t*. Errors (if any) have to be handled in the event loop.

If you want to handle errors directly with *xcb_request_check* instead, use *xcb_send_event_checked*. See **xcb-requests(3)** for details.

ERRORS

xcb_value_error_t

The given *event* is neither a core event nor an event defined by an extension.

xcb_window_error_t

The specified *destination* window does not exist.

EXAMPLE

```
/*
 * Tell the given window that it was configured to a size of 800x600 pixels.
 *
 */
void my_example(xcb_connection_t *conn, xcb_window_t window) {
   /* Every X11 event is 32 bytes long. Therefore, XCB will copy 32 bytes.
    * In order to properly initialize these bytes, we allocate 32 bytes even
    * though we only need less for an xcb_configure_notify_event_t */
   xcb_configure_notify_event_t *event = calloc(32, 1);

   event->event = window;
   event->window = window;
   event->response_type = XCB_CONFIGURE_NOTIFY;

   event->x = 0;
   event->y = 0;
   event->width = 800;
   event->height = 600;

   event->border_width = 0;
   event->above_sibling = XCB_NONE;
   event->override_redirect = false;
```

```
    xcb_send_event(conn, false, window, XCB_EVENT_MASK_STRUCTURE_NOTIFY,
                   (char*)event);
    xcb_flush(conn);
    free(event);
}
```

SEE ALSO

xcb-requests(3), **xcb-examples**(3), **xcb_configure_notify_event_t**(3)

AUTHOR

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