NAME

XSendEvent, XDisplayMotionBufferSize, XGetMotionEvents, XTimeCoord - send events and pointer motion history structure

SYNTAX

Status XSendEvent(Display **display*, Window *w*, Bool *propagate*, long *event_mask*, XEvent **event_send*);

unsigned long XDisplayMotionBufferSize(Display **display*);

XTimeCoord *XGetMotionEvents(Display **display*, Window *w*, Time *start*, Time *stop*, int **nevents_return*);

ARGUMENTS

display Specifies the connection to the X server.

event_mask Specifies the event mask.

event_send Specifies the event that is to be sent.

nevents_return Returns the number of events from the motion history buffer.

propagate Specifies a Boolean value.

start

stop Specify the time interval in which the events are returned from the motion history buffer. You can pass a timestamp or **CurrentTime**. **PointerWindow**, or **InputFocus**.

w Specifies the window the window the event is to be sent to.

DESCRIPTION

The **XSendEvent** function identifies the destination window, determines which clients should receive the specified events, and ignores any active grabs. This function requires you to pass an event mask. For a discussion of the valid event mask names, see section 10.3. This function uses the w argument to identify the destination window as follows:

- If w is **PointerWindow**, the destination window is the window that contains the pointer.
- If w is **InputFocus** and if the focus window contains the pointer, the destination window is the

window that contains the pointer; otherwise, the destination window is the focus window.

To determine which clients should receive the specified events, **XSendEvent** uses the propagate argument as follows:

- If event_mask is the empty set, the event is sent to the client that created the destination window. If that client no longer exists, no event is sent.

- If propagate is **False**, the event is sent to every client selecting on destination any of the event types in the event_mask argument.

- If propagate is **True** and no clients have selected on destination any of the event types in event-mask, 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.

The event in the **XEvent** structure must be one of the core events or one of the events defined by an extension (or a **BadValue** error results) so that the X server can correctly byte-swap the contents as necessary. The contents of the event are otherwise unaltered and unchecked by the X server except to force send_event to **True** in the forwarded event and to set the serial number in the event correctly; therefore these fields and the display field are ignored by **XSendEvent**.

XSendEvent returns zero if the conversion to wire protocol format failed and returns nonzero otherwise. **XSendEvent** can generate **BadValue** and **BadWindow** errors.

The server may retain the recent history of the pointer motion and do so to a finer granularity than is reported by **MotionNotify** events. The **XGetMotionEvents** function makes this history available.

The **XGetMotionEvents** function returns all events in the motion history buffer that fall between the specified start and stop times, inclusive, and that have coordinates that lie within the specified window (including its borders) at its present placement. If the server does not support motion history, if the start time is later than the stop time, or if the start time is in the future, no events are returned; **XGetMotionEvents** returns NULL. If the stop time is in the future, it is equivalent to specifying **CurrentTime**. **XGetMotionEvents** can generate a **BadWindow** error.

STRUCTURES

The **XTimeCoord** structure contains:

```
typedef struct {
    Time time;
    short x, y;
} XTimeCoord;
```

The time member is set to the time, in milliseconds. The x and y members are set to the coordinates of the pointer and are reported relative to the origin of the specified window.

DIAGNOSTICS

BadValue Some numeric value falls outside the range of values accepted by the request. Unless a specific range is specified for an argument, the full range defined by the argument's type is accepted. Any argument defined as a set of alternatives can generate this error.

BadWindow A value for a Window argument does not name a defined Window.

SEE ALSO

XAnyEvent(3), XIfEvent(3), XNextEvent(3), XPutBackEvent(3)
Xlib - C Language X Interface