XSendExtensionEvent – send input extension events to a client

Status XSendExtensionEvent(*display***,** *device***,** *destination***,** *propagate***,** *event_count***,** *event_list***,** *event_send***)**

Display ****display*;**
XDevice ****device*;**
Window *destination***;**
Bool *propagate***;**
int *event_count***;**
XEventClass ****event_list*;**
XEvent ****event_send*;**

display **Specifies the connection to the X server.** *device* **Specifies the device from which the events are to be sent.** *destination* **Specifies the window the event is to be sent to. You can pass window id,** *PointerWindow***, or** *InputFocus***.** *propagate* **Specifies a Boolean value that is either True or False.** *event_count* **Specifies the count of XEventClasses in event_list.** *event_list* **Specifies the list of event selections to be used.** *event_send* **Specifies a pointer to the event that is to be sent.**

The *XSendExtensionEvent* **request identifies the destination window, determines which clients should receive the specified events, and ignores any active grabs. This request requires you to pass an event class list. For a discussion of the valid event class names, see** *XOpenDevice(3X11)***. This request 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, *XSendExtensionEvent* uses the propagate argument as follows:

- If event_list 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 specified by the event_list array.
- If propagate is *True* and no clients have selected on destination any of the events specified by the event_list array, the destination is replaced with the closest ancestor of destination for which some client has selected a type specified by the event-list array 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 events specified in the event_list array.

The event in the *XEvent* structure must be one of the events defined by the input 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.

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

BadDevice **An invalid device was specified. The specified device does not exist or has not been opened by this client via** *XOpenInputDevice***. This error may also occur if the specified device is the X keyboard or X pointer device.** *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.**