

XSelectExtensionEvent, XGetSelectedExtensionEvents – select extension events, get the list of currently selected extension events

XSelectExtensionEvent(*display, w, event_list, event_count*)

Display **display*;
Window *w*;
XEventClass **event_list*;
int *event_count*;

XGetSelectedExtensionEvents(*display, w, this_client_event_count_return,*
this_client_event_list_return, all_clients_event_count_return, all_clients_event_list_return)

Display **display*;
Window *w*;
int *this_client_event_count_return*;
XEventClass **this_client_event_list_return*;
int *all_clients_event_count_return*;
XEventClass **all_clients_event_list_return*;

display Specifies the connection to the X server. *w* Specifies the window whose events you are interested in. *event_list* Specifies the list of event classes that describe the events you are interested in. *event_count* Specifies the count of event classes in the event list. *this_client_event_count_return* Returns the count of event classes selected by this client. *this_client_event_list_return* Returns a pointer to the list of event classes selected by this client. *all_clients_event_count_return* Returns the count of event classes selected by all clients. *all_clients_event_list_return* Returns a pointer to the list of event classes selected by all clients.

The *XSelectExtensionEvent* request causes the X server to report the events associated with the specified list of event classes. Initially, X will not report any of these events. Events are reported relative to a window. If a window is not interested in a device event, it usually propagates to the closest ancestor that is interested, unless the *do_not_propagate* mask prohibits it.

Multiple clients can select for the same events on the same window with the following restrictions:

- Multiple clients can select events on the same window because their event masks are disjoint. When the X server generates an event, it reports it to all interested clients.
- Only one client at a time can select a *DeviceButtonPress* event with automatic passive grabbing enabled, which is associated with the event class *DeviceButtonPressGrab*. To receive *DeviceButtonPress* events without automatic passive grabbing, use event class *DeviceButtonPress* but do not specify event class *DeviceButtonPressGrab*.

The server reports the event to all interested clients.

Information contained in the *XDevice* structure returned by *XOpenDevice* is used by macros to obtain the event classes that clients use in making *XSelectExtensionEvent* requests. Currently defined macros include *DeviceKeyPress*, *DeviceKeyRelease*, *DeviceButtonPress*, *DeviceButtonRelease*, *DeviceMotionNotify*, *DeviceFocusIn*, *DeviceFocusOut*, *ProximityIn*, *ProximityOut*, *DeviceStateNotify*, *DeviceMappingNotify*, *ChangeDeviceNotify*, *DevicePointerMotionHint*, *DeviceButton1Motion*, *DeviceButton2Motion*, *DeviceButton3Motion*, *DeviceButton4Motion*, *DeviceButton5Motion*, *DeviceButtonMotion*, *DeviceOwnerGrabButton*, *DeviceButtonPressGrab*, and *NoExtensionEvent*.

To obtain the proper event class for a particular device, one of the above macros is invoked using the *XDevice* structure for that device. For example,

```
DeviceKeyPress (*device, type, eventclass);
```

returns the *DeviceKeyPress* event type and the eventclass for selecting *DeviceKeyPress* events from this device.

XSelectExtensionEvent can generate a *BadWindow* or *BadClass* error. The *XGetSelectedExtensionEvents* request reports the extension events selected by this client and all clients for the specified window. This request returns pointers to two *XEventClass* arrays. One lists the input extension events selected by this client from the specified window. The other lists the event classes selected by all clients from the specified window. You should use *XFree* to free these two arrays.

XGetSelectedExtensionEvents can generate a *BadWindow* error.

***BadWindow* A value for a Window argument does not name a defined Window. *BadClass* A value for an XEventClass argument is invalid.**

Programming with Xlib