

NAME

XkbSelectEvents - Selects and / or deselects for delivery of one or more Xkb events and has them delivered under all conditions

SYNOPSIS

Bool XkbSelectEvents (Display *display, unsigned int device_spec, unsigned long int bits_to_change, unsigned long int values_for_bits);

ARGUMENTS

display

connection to the X server

device_spec

device ID, or XkbUseCoreKbd

bits_to_change

determines events to be selected / deselected

values_for_bits

1=>select, 0->deselect; for events in bits_to_change

DESCRIPTION

Xkb events are selected using an event mask, much the same as normal core X events are selected. However, unlike selecting core X events, where you must specify the selection status (on or off) for all possible event types whenever you wish to change the selection criteria for any one event, Xkb allows you to restrict the specification to only the event types you wish to change. This means that you do not need to remember the event selection values for all possible types each time you want to change one of them.

Many Xkb event types are generated under several different circumstances. When selecting to receive an Xkb event, you may specify either that you want it delivered under all circumstances, or that you want it delivered only for a subset of the possible circumstances.

You can also deselect an event type that was previously selected for, using the same granularity.

Xkb provides two functions to select and deselect delivery of Xkb events. *XkbSelectEvents* allows you to select or deselect delivery of more than one Xkb event type at once. Events selected using *XkbSelectEvents* are delivered to your program under all circumstances that generate the events. To restrict delivery of an event to a subset of the conditions under which it occurs, use *XkbSelectEventDetails*. *XkbSelectEventDetails* only allows you to change the selection conditions for

a single event at a time, but it provides a means of fine-tuning the conditions under which the event is delivered.

This request changes the Xkb event selection mask for the keyboard specified by *device_spec*.

Each Xkb event that can be selected is represented by a bit in the *bits_to_change* and *values_for_bits* masks. Only the event selection bits specified by the *bits_to_change* parameter are affected; any unspecified bits are left unchanged. To turn on event selection for an event, set the bit for the event in the *bits_to_change* parameter and set the corresponding bit in the *values_for_bits* parameter. To turn off event selection for an event, set the bit for the event in the *bits_to_change* parameter and do not set the corresponding bit in the *values_for_bits* parameter. The valid values for both of these parameters are an inclusive bitwise OR of the masks shown in Table 1. There is no interface to return your client's current event selection mask. Clients cannot set other clients' event selection masks.

The X server reports the events defined by Xkb to your client application only if you have requested them via a call to *XkbSelectEvents* or *XkbSelectEventDetails*. Specify the event types in which you are interested in a mask.

Table 1 lists the event mask constants that can be specified with the *XkbSelectEvents* request and the circumstances in which the mask should be specified.

Table 1 XkbSelectEvents Mask

Constants

Event Mask	Value	Notification Wanted
XkbNewKeyboardNotifyMask	(1L<<0)	Keyboard geometry change
XkbMapNotifyMask	(1L<<1)	Keyboard mapping change
XkbStateNotifyMask	(1L<<2)	Keyboard state change
XkbControlsNotifyMask	(1L<<3)	Keyboard control change
XkbIndicatorStateNotifyMask	(1L<<4)	Keyboard indicator state change
XkbIndicatorMapNotifyMask	(1L<<5)	Keyboard indicator map change
XkbNamesNotifyMask	(1L<<6)	Keyboard name

		change
XkbCompatMapNotifyMask	(1L<<7)	Keyboard compat map change
XkbBellNotifyMask	(1L<<8)	Bell
XkbActionMessageMask	(1L<<9)	Action message
XkbAccessXNotifyMask	(1L<<10)	AccessX features
XkbExtensionDeviceNotifyMask	(1L<<11)	Extension device
XkbAllEventsMask	(0xFFF)	All Xkb events

If a bit is not set in the *bits_to_change* parameter, but the corresponding bit is set in the *values_for_bits* parameter, a BadMatch protocol error results. If an undefined bit is set in either the *bits_to_change* or the *values_for_bits* parameter, a BadValue protocol error results.

All event selection bits are initially zero for clients using the Xkb extension. Once you set some bits, they remain set for your client until you clear them via another call to *XkbSelectEvents*.

XkbSelectEvents returns False if the Xkb extension has not been initialized and True otherwise.

RETURN VALUES

True	The <i>XkbSelectEvents</i> function returns True if the Xkb extension has been initialized.
False	The <i>XkbSelectEvents</i> function returns False if the Xkb extension has not been initialized.

SEE ALSO

XkbSelectEventDetails(3), **XkbUseCoreKbd(3)**