

**NAME**

XkbSetControls - Copies changes to the X server based on a modified *ctrls* structure in a local copy of the keyboard description

**SYNOPSIS**

**Bool XkbSetControls (Display \**display*, unsigned long *which*, XkbDescPtr *xkb*);**

**ARGUMENTS**

*display*

connection to X server

*which*

mask of controls requested

*xkb* keyboard description for controls information

**DESCRIPTION**

For each bit that is set in the *which* parameter, *XkbSetControls* sends the corresponding values from the *xkb->ctrls* field to the server. Valid values for *which* are any combination of the masks listed in Table 1 that have "ok" in the *which* column.

Table 1 shows the actual values for the individual mask bits used to select controls for modification and to enable and disable the control. Note that the same mask bit is used to specify general modifications to the parameters used to configure the control (*which*), and to enable and disable the control (*enabled\_ctrls*). The anomalies in the table (no "ok" in column) are for controls that have no configurable attributes; and for controls that are not boolean controls and therefore cannot be enabled or disabled.

Table 1 Controls Mask

Bits

Mask	which	enabledValue	
Bit	or	changed_ctrls_ctrls	
XkbRepeatKeysMask	ok	ok	(1L<<0)
XkbSlowKeysMask	ok	ok	(1L<<1)
XkbBounceKeysMask	ok	ok	(1L<<2)
XkbStickyKeysMask	ok	ok	(1L<<3)

XkbMouseKeysMask	ok	ok	(1L<<4)
XkbMouseKeysAccelMask	ok	ok	(1L<<5)
XkbAccessXKeysMask	ok	ok	(1L<<6)
XkbAccessXTimeoutMask	ok	ok	(1L<<7)
XkbAccessXFeedbackMask	ok	ok	(1L<<8)
XkbAudibleBellMask		ok	(1L<<9)
XkbOverlay1Mask		ok	(1L<<10)
XkbOverlay2Mask		ok	(1L<<11)
XkbIgnoreGroupLockMask		ok	(1L<<12)
XkbGroupsWrapMask	ok		(1L<<27)
XkbInternalModsMask	ok		(1L<<28)
XkbIgnoreLockModsMask	ok		(1L<<29)
XkbPerKeyRepeatMask	ok		(1L<<30)
XkbControlsEnabledMask	ok		(1L<<31)
XkbAccessXOptionsMask	ok	ok	(XkbStickyKeysMask   XkbAccessXFeedbackMask)
XkbAllBooleanCtrlsMask		ok	(0x00001FFF)
XkbAllControlsMask	ok		(0xF8001FFF)

If *xkb->ctrls* is NULL, the server does not support a compatible version of Xkb, or the Xkb extension has not been properly initialized, *XkbSetControls* returns False. Otherwise, it sends the request to the X server and returns True.

Note that changes to attributes of controls in the XkbControlsRec structure are apparent only when the associated control is enabled, although the corresponding values are still updated in the X server. For example, the *repeat\_delay* and *repeat\_interval* fields are ignored unless the RepeatKeys control is enabled (that is, the X server's equivalent of *xkb->ctrls* has XkbRepeatKeyMask set in *enabled\_ctrls*). It is permissible to modify the attributes of a control in one call to *XkbSetControls* and enable the control in a subsequent call. See *XkbChangeEnabledControls* for more information on enabling and disabling controls.

Note that the *enabled\_ctrls* field is itself a control - the EnabledControls control. As such, to set a specific configuration of enabled and disabled boolean controls, you must set *enabled\_ctrls* to the appropriate bits to enable only the controls you want and disable all others, then specify the XkbControlsEnabledMask in a call to *XkbSetControls*.

Because this is somewhat awkward if all you want to do is enable and disable controls, and not modify any of their attributes, a convenience function is also provided for this purpose, *XkbChangeEnabledControls*.

**RETURN VALUES**

- True**                The *XkbSetControls* function returns True when it sends the request to the X server.
- False**              The *XkbSetControls* function returns False when *xkb->ctrls* is NULL, the server does not support a compatible version of Xkb, or the Xkb extension has not been properly initialized.

**DIAGNOSTICS**

- BadAlloc**            Unable to allocate storage
- BadMatch**          A compatible version of Xkb was not available in the server or an argument has correct type and range, but is otherwise invalid

**SEE ALSO**

**XkbChangeEnabledControls(3), XkbFreeControls(3)**