

**NAME**

XkbGetControls - Finds the current state of Xkb server controls

**SYNOPSIS**

**Status XkbGetControls (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**

*XkbGetControls* queries the server for the requested control information, waits for a reply, and then copies the server's values for the requested information into the *ctrls* structure of the *xkb* argument. Only those components specified by the *which* parameter are copied. 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 Table 1 (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	enabled	Value
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, *XkbGetControls* allocates and initializes it before obtaining the values specified by *which*. If *xkb->ctrls* is not NULL, *XkbGetControls* modifies only those portions of *xkb->ctrls* corresponding to the values specified by *which*.

*XkbGetControls* returns Success if successful; otherwise, it returns BadAlloc if it cannot obtain sufficient storage, BadMatch if *xkb* is NULL or *which* is empty, or BadImplementation.

To free the *ctrls* member of a keyboard description, use *XkbFreeControls*.

The *num\_groups* field in the *ctrls* structure is always filled in by *XkbGetControls*, regardless of which bits are selected by *which*.

## RETURN VALUES

Success            The *XkbGetControls* function returns Success if successful.

## DIAGNOSTICS

**BadAlloc**            Unable to allocate storage

### **BadImplementation**

Invalid reply from server

**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**

**XkbFreeControls(3)**