NAME

XkbChangeControls - Provides a flexible method for updating the controls in a server to match those in the changed keyboard description

SYNOPSIS

Bool XkbChangeControls (Display ****dpy*, XkbDescPtr** *xkb***, XkbControlsChangesPtr** *changes***);**

ARGUMENTS

dpy connection to X server

xkb keyboard description with changed xkb->ctrls

changes
which parts of xkb->ctrls have changed

DESCRIPTION

The XkbControlsChangesRec structure allows applications to track modifications to an XkbControlsRec structure and thereby reduce the amount of traffic sent to the server. The same XkbControlsChangesRec structure may be used in several successive modifications to the same XkbControlsRec structure, then subsequently used to cause all of the changes, and only the changes, to be propagated to the server.

The *changed_ctrls* field is a mask specifying which logical sets of data in the controls structure have been modified. In this context, modified means *set,* that is, if a value is set to the same value it previously contained, it has still been modified, and is noted as changed. Valid values for *changed_ctrls* are any combination of the masks listed in Table 1 that have "ok" in the *changed_ctrls* column. Setting a bit implies the corresponding data fields from the "Relevant XkbControlsRec Data Fields" column in Table 1 have been modified. The *enabled_ctrls_changes* field specifies which bits in the *enabled_ctrls* field have changed. If the number of keyboard groups has changed, the *num_groups_changed* field is set to True.

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 Xkb

Controls

Control	Control Selection Mask (which parameter)	Relevant XkbControlsRec DataFields	Boolean Control enabled_ctrls bit	Section
AccessXFeedback	XkbAccessXFeedbackMask	ax_options: XkbAX_*FBMask	XkbAccessXFeedbackMask	10.6.3
AccessXKeys			XkbAccessXKeysMask	10.6.1
AccessXTimeout	XkbAccessXTimeoutMask	ax_timeout axt_opts_mask axt_opts_values axt_ctrls_mask axt_ctrls_values	XkbAccessXTimeoutMask	10.6.2
AudibleBell			XkbAudibleBellMask	9.2
AutoReset				10.1.2
BounceKeys	XkbBounceKeysMask	debounce_delay	XkbBounceKeysMask	10.6.7
Detectable-Autorepeat				10.3.3
EnabledControls	XkbControlsEnabledMask	enabled_ctrls	Non-Boolean Control	10.1.1
GroupsWrap	XkbGroupsWrapMask	groups_wrap	Non-Boolean Control	10.7.1
IgnoreGroupLock			XkbIgnoreGroupLockMask	10.7.3
IgnoreLockMods	XkbIgnoreLockModsMask	ignore_lock	Non-Boolean Control	5.1
InternalMods	XkbInternalModsMask	internal	Non-Boolean Control	5.1
MouseKeys	XkbMouseKeysMask	mk_dflt_btn	XkbMouseKeysMask	10.5.1
MouseKeysAccel	XkbMouseKeysAccelMask	mk_delay mk_interval mk_time_to_max mk_max_speed mk_curve	XkbMouseKeysAccelMask	10.5.2
Overlay1			XkbOverlay1Mask	10.4
Overlay2			XkbOverlay2Mask	10.4
PerKeyRepeat	XkbPerKeyRepeatMask	per_key_repeat	Non-Boolean Control	10.3.1
RepeatKeys	XkbRepeatKeysMask	repeat_delay	XkbRepeatKeysMask repeat_interval	10.3

SlowKeys	XkbSlowKeysMask	slow_keys_delay	XkbSlowKeysMask	10.6.6
StickyKeys	XkbStickyKeysMask	ax_options: XkbAX_TwoKeysMask XkbAX_LatchToLockMask	XkbStickyKeysMask	10.6.8

Table 2 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 2 Controls Mask Bits

Mask Bit	which or changed_ctrls	enabled_ctrls	Value
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 you have an Xkb description with controls that have been modified and an XkbControlsChangesRec that describes the changes that have been made, the *XkbChangeControls* function provides a flexible method for updating the controls in a server to match those in the changed keyboard description.

XkbChangeControls copies any controls fields specified by *changes* from the keyboard description controls structure, *xkb->ctrls*, to the server specified by *dpy*.

STRUCTURES

The XkbControlsChangesRec structure is defined as follows:

```
typedef struct _XkbControlsChanges {
   unsigned int  changed_ctrls;         /* bits indicating changed control data */
   unsigned int  enabled_ctrls_changes; /* bits indicating enabled/disabled controls */
   Bool          num_groups_changed;    /* True if number of keyboard groups changed */
} XkbControlsChangesRec,*XkbControlsChangesPtr;
```

SEE ALSO

XkbChangeControls(3), XkbChangeDeviceInfo(3), XkbChangeEnabledControls(3), XkbChangeMap(3), XkbChangeNames(3), XkbChangeTypesOfKey(3)