NAME

XkbGetDeviceInfo - Determine whether the X server allows Xkb access to particular capabilities of input devices other than the core X keyboard, or to determine the status of indicator maps, indicator names or button actions on a non-KeyClass extension device

SYNOPSIS

XkbDeviceInfoPtr XkbGetDeviceInfo (Display ****dpy*, unsigned int** *which***, unsigned int** *device_spec***, unsigned int** *ind_class***, unsigned int** *ind_id***);**

ARGUMENTS

dpy connection to X server

which
mask indicating information to return

device_spec
device ID, or XkbUseCoreKbd

ind_class
feedback class for indicator requests

ind_id
feedback ID for indicator requests

DESCRIPTION

To determine whether the X server allows Xkb access to particular capabilities of input devices other than the core X keyboard, or to determine the status of indicator maps, indicator names or button actions on a non-KeyClass extension device, use *XkbGetDeviceInfo.*

XkbGetDeviceInfo returns information about the input device specified by *device_spec.* Unlike the *device_spec* parameter of most Xkb functions, *device_spec* does not need to be a keyboard device. It must, however, indicate either the core keyboard or a valid X Input Extension device.

The *which* parameter is a mask specifying optional information to be returned. It is an inclusive OR of one or more of the values from Table 1 and causes the returned XkbDeviceInfoRec to contain values for the corresponding fields specified in the table.

Table 1 XkbDeviceInfoRec Mask Bits

Name	XkbDeviceInfoRec Value	Capability If Set

	Fields Effected		
XkbXI_KeyboardsMask		(1L <<0)	Clients can use all Xkb requests and events with KeyClass devices supported by the input device extension.
XkbXI_ButtonActionsMask	num_btns btn_acts	(1L <<1)	Clients can assign key actions to buttons non-KeyClass input extension devices.
XkbXI_IndicatorNamesMask	leds->names	(1L <<2)	Clients can assign names to indicators on non-KeyClass input extension devices.
XkbXI_IndicatorMapsMask	leds->maps	(1L <<3)	Clients can assign indicator maps to indicators on non-KeyClass input extension devices.
XkbXI_IndicatorStateMask	leds->state	(1L <<4)	Clients can request the status of indicators on non-KeyClass input extension devices.
XkbXI_IndicatorsMask	sz_leds num_leds leds->*	(0x1c)	XkbXI_IndicatorNamesMask | XkbXI_IndicatorMapsMask | XkbXI_IndicatorStateMask
XkbXI_UnsupportedFeaturesMask	unsupported	(1L <<15)	
XkbXI_AllDeviceFeaturesMask	Those selected by Value Column masks	(0x1e)	XkbXI_IndicatorsMask | XkbSI_ButtonActionsMask
XkbXI_AllFeaturesMask	Those selected	(0x1f)	XkbSI_AllDeviceFeaturesMask |

	by Value Column masks		XkbSI_KeyboardsMask
XkbXI_AllDetailsMask	Those selected by Value column masks	(0x801f)	XkbXI_AllFeaturesMask | XkbXI_UnsupportedFeaturesMask

The XkbDeviceInfoRec returned by *XkbGetDeviceInfo* always has values for *name* (may be a null string, ""), *type, supported, unsupported, has_own_state, dflt_kbd_fd,* and *dflt_kbd_fb.* Other fields are filled in as specified by *which.*

Upon return, the *supported* field will be set to the inclusive OR of zero or more bits from Table 1; each bit set indicates an optional Xkb extension device feature supported by the server implementation, and a client may modify the associated behavior.

If the XkbButtonActionsMask bit is set in *which,* the XkbDeviceInfoRec returned will have the button actions *(btn_acts* field) filled in for all buttons.

If *which* includes one of the bits in XkbXI_IndicatorsMask, the feedback class of the indicators must be specified in *ind_class,* and the feedback ID of the indicators must be specified in *ind_id.* If the request does not include any of the bits in XkbXI_IndicatorsMask, the *ind_class* and *ind_id* parameters are ignored. The class and ID can be obtained via the input device extension *XListInputDevices* request.

If any of the XkbXI_IndicatorsMask bits are set in *which,* the XkbDeviceInfoRec returned will have filled in the portions of the *leds* structure corresponding to the indicator feedback identified by *ind_class* and *ind_id.* The *leds* vector of the XkbDeviceInfoRec is allocated if necessary and *sz_leds* and *num_leds* filled in. The *led_class, led_id* and *phys_indicators* fields of the *leds* entry corresponding to *ind_class* and *ind_id* are always filled in. If *which* contains XkbXI_IndicatorNamesMask, the *names_present* and *names* fields of the *leds* structure corresponding to *ind_class* and *ind_id* are returned. If *which* contains XkbXI_IndicatorStateMask, the corresponding *state* field is updated. If *which* contains XkbXI_IndicatorMapsMask, the *maps_present* and *maps* fields are updated.

Xkb provides convenience functions to request subsets of the information available via *XkbGetDeviceInfo.* These convenience functions mirror some of the mask bits. The functions all take an XkbDeviceInfoPtr as an input argument and operate on the X Input Extension device specified by the *device_spec* field of the structure. Only the parts of the structure indicated in the function description are updated. The XkbDeviceInfoRec structure used in the function call can be obtained by calling *XkbGetDeviceInfo* or can be allocated by calling *XkbAllocDeviceInfo.*

STRUCTURES

Information about X Input Extension devices is transferred between a client program and the Xkb extension in an XkbDeviceInfoRec structure:

```
typedef struct {
    char *              name;          /* name for device */
    Atom                type;          /* name for class of devices */
    unsigned short      device_spec;   /* device of interest */
    Bool                has_own_state; /* True=>this device has its own state */
    unsigned short      supported;     /* bits indicating supported capabilities */
    unsigned short      unsupported;   /* bits indicating unsupported capabilities */
    unsigned short      num_btns;      /* number of entries in btn_acts */
    XkbAction *         btn_acts;      /* button actions */
    unsigned short      sz_leds;       /* total number of entries in LEDs vector */
    unsigned short      num_leds;      /* number of valid entries in LEDs vector */
    unsigned short      dflt_kbd_fb;   /* input extension ID of default (core kbd) indicator */
    unsigned short      dflt_led_fb;   /* input extension ID of default indicator feedback */
    XkbDeviceLedInfoPtr  leds;         /* LED descriptions */
} XkbDeviceInfoRec, *XkbDeviceInfoPtr;
```

SEE ALSO

XkbAllocDeviceInfo(3), **XListInputDevices**(3)