NAME

XkbGetDeviceLedInfo - Query the indicator names, maps, and state associated with an LED feedback of an input extension device

SYNOPSIS

Status XkbGetDeviceLedInfo (Display ****dpy*, XkbDeviceInfoPtr** *device_info***, unsigned int** *led_class***, unsigned int** *led_id***, unsigned int** *which***);**

ARGUMENTS

dpy connection to X server

device_info
structure to update with results

led_class
LED feedback class assigned by input extension

led_id
LED feedback ID assigned by input extension

which
mask indicating desired information

DESCRIPTION

XkbGetDeviceLedInfo queries the server for the desired LED information for the feedback specified by *led_class* and *led_id* for the X input extension device indicated by *device_spec*->*device_info* and waits for a reply. If successful, *XkbGetDeviceLedInfo* backfills the relevant fields of *device_info* as determined by *which* with the results and returns Success. Valid values for *which* are the inclusive OR of any of XkbXI_IndicatorNamesMask, XkbXI_IndicatorMapsMask, and XkbXI_IndicatorStateMask.

The fields of *device_info* that are filled in when this request succeeds are *name, type, supported,* and *unsupported,* and portions of the *leds* structure corresponding to *led_class* and *led_id* as indicated by the bits set in *which.* The *device_info*->*leds* vector is allocated if necessary and *sz_leds* and *num_leds* filled in. The *led_class, led_id* and *phys_indicators* fields of the *device_info*->*leds* entry corresponding to *led_class* and *led_id* are always filled in.

If *which* contains XkbXI_IndicatorNamesMask, the *names_present* and *names* fields of the *device_info*->*leds* structure corresponding to *led_class* and *led_id* are updated, if *which* contains XkbXI_IndicatorStateMask, the corresponding *state* field is updated, and if *which* contains XkbXI_IndicatorMapsMask, the *maps_present* and *maps* fields are updated.

If a compatible version of Xkb is not available in the server or the Xkb extension has not been properly initialized, *XkbGetDeviceLedInfo* returns BadAccess. If allocation errors occur, a BadAlloc status is returned. If the device has no indicators, a BadMatch error is returned. If *ledClass* or *ledID* have illegal values, a BadValue error is returned. If they have legal values but do not specify a feedback that contains LEDs and is associated with the specified device, a BadMatch error is returned.

DIAGNOSTICS

BadAccess The Xkb extension has not been properly initialized

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

BadValue An argument is out of range