

XGetDeviceKeyMapping, XChangeDeviceKeyMapping – query or change device key mappings

XChangeDeviceKeyMapping(*display*, *device*, *first_keycode*, *keysyms_per_keycode*, *keysyms*, *keycode_count*)

```
Display *display;  
XDevice *device;  
int first_keycode;  
int keysyms_per_keycode;  
KeySym *keysyms;  
int keycode_count;
```

KeySym ***XGetDeviceKeyMapping**(*display*, *device*, *first_keycode*, *keycode_count*, *keysyms_per_keycode_return*)

```
Display *display;  
XDevice *device;  
KeyCode first_keycode;  
int keycode_count;  
int *keysyms_per_keycode_return;
```

display Specifies the connection to the X server. *device* Specifies the device whose key mapping is to be queried or modified. *first_keycode* Specifies the first KeyCode to be returned. *keycode_count* Specifies the number of KeyCodes to be returned or modified. *keysyms_per_keycode* Specifies the number of KeySyms per KeyCode. *keysyms_per_keycode_return* Specifies the address of a variable into which the number of KeySyms per KeyCode will be returned. *keysyms* Specifies the address of an array of KeySyms.

For the specified device, the *XGetDeviceKeyMapping* request returns the symbols for the specified number of KeyCodes starting with *first_keycode*. The value specified in *first_keycode* must be greater than or equal to *min_keycode* as returned by *XListInputDevices*, or a *BadValue* error results. In addition, the following expression must be less than or equal to *max_keycode* as returned by *XListInputDevices*:

$$\text{first_keycode} + \text{keycode_count} - 1$$

If this is not the case, a *BadValue* error results. The number of elements in the KeySyms list is:

$$\text{keycode_count} * \text{keysyms_per_keycode_return}$$

KeySym number *N*, counting from zero, for KeyCode *K* has the following index in the list, counting from zero:

$$(\text{K} - \text{first_code}) * \text{keysyms_per_code_return} + \text{N}$$

The X server arbitrarily chooses the *keysyms_per_keycode_return* value to be large enough to report all requested symbols. A special KeySym value of *NoSymbol* is used to fill in unused elements for individual KeyCodes. To free the storage returned by *XGetDeviceKeyMapping*, use *XFree*.

If the specified device does not support input class keys, a *BadMatch* error will result.

XGetDeviceKeyMapping can generate a *BadDevice*, *BadMatch*, or *BadValue* error.

For the specified device, the *XChangeDeviceKeyMapping* request defines the symbols for the specified number of KeyCodes starting with *first_keycode*. The symbols for KeyCodes outside this range remain unchanged. The number of elements in *keysyms* must be:

$\text{num_codes} * \text{keysyms_per_keycode}$

The specified `first_keycode` must be greater than or equal to `min_keycode` returned by *XListInputDevices*, or a *BadValue* error results. In addition, the following expression must be less than or equal to `max_keycode` as returned by *XListInputDevices*, or a *BadValue* error results:

$\text{first_keycode} + \text{num_codes} - 1$

KeySym number N, counting from zero, for KeyCode K has the following index in `keysyms`, counting from zero:

$(K - \text{first_keycode}) * \text{keysyms_per_keycode} + N$

The specified `keysyms_per_keycode` can be chosen arbitrarily by the client to be large enough to hold all desired symbols. A special KeySym value of *NoSymbol* should be used to fill in unused elements for individual KeyCodes. It is legal for *NoSymbol* to appear in nontrailing positions of the effective list for a KeyCode. *XChangeDeviceKeyMapping* generates a *DeviceMappingNotify* event that is sent to all clients that have selected that type of event.

There is no requirement that the X server interpret this mapping. It is merely stored for reading and writing by clients.

If the specified device does not support input class keys, a *BadMatch* error results.

XChangeDeviceKeyMapping can generate a *BadDevice*, *BadMatch*, *BadAlloc*, or *BadValue* error.

BadDevice An invalid device was specified. The specified device does not exist or has not been opened by this client via *XOpenInputDevice*. This error may also occur if the specified device is the X keyboard or X pointer device. ***BadMatch*** This error may occur if an *XGetDeviceKeyMapping* or *XChangeDeviceKeyMapping* request was made specifying a device that has no keys. ***BadValue*** Some numeric value falls outside the range of values accepted by the request. Unless a specific range is specified for an argument, the full range defined by the argument's type is accepted. Any argument defined as a set of alternatives can generate this error. ***BadAlloc*** The server failed to allocate the requested resource or server memory.

XSetDeviceButtonMapping(3X11)

XSetDeviceModifierMapping(3X11)

Programming with Xlib