

## XGetDeviceModifierMapping, XSetDeviceModifierMapping – query or change device modifier mappings

**XSetDeviceModifierMapping**(*display*, *device*, *modmap*)

```
Display *display;  
XDevice *device;  
XModifierKeymap *modmap;
```

**XModifierKeymap \*XGetDeviceModifierMapping**(*display*, *device*)

```
Display *display;  
XDevice *device;
```

*display* Specifies the connection to the X server. *device* Specifies the device whose modifier mapping is to be queried or modified. *modmap* Specifies a pointer to the *XModifierKeymap* structure.

The *XSetDeviceModifierMapping* request specifies the *KeyCodes* of the keys (if any) that are to be used as modifiers for the specified device. If it succeeds, the X server generates a *DeviceMappingNotify* event, and *XSetDeviceModifierMapping* returns *MappingSuccess*. X permits at most eight modifier keys. If more than eight are specified in the *XModifierKeymap* structure, a *BadLength* error results.

The *modifiermap* member of the *XModifierKeymap* structure contains eight sets of *max\_keypermod* *KeyCodes*, one for each modifier in the order *Shift*, *Lock*, *Control*, *Mod1*, *Mod2*, *Mod3*, *Mod4*, and *Mod5*. Only nonzero *KeyCodes* have meaning in each set, and zero *KeyCodes* are ignored. In addition, all of the nonzero *KeyCodes* must be in the range specified by *min\_keycode* and *max\_keycode* as returned by *XLis-InputDevices*, or a *BadValue* error results. No *KeyCode* may appear twice in the entire map, or a *BadValue* error results.

An X server can impose restrictions on how modifiers can be changed, for example, if certain keys do not generate up transitions in hardware, if auto-repeat cannot be disabled on certain keys, or if multiple modifier keys are not supported. If some such restriction is violated, the status reply is *MappingFailed*, and none of the modifiers are changed. If the new *KeyCodes* specified for a modifier differ from those currently defined and any (current or new) keys for that modifier are in the logically down state, *XSetDeviceModifierMapping* returns *MappingBusy*, and none of the modifiers is changed.

*XSetDeviceModifierMapping* can generate *BadLengthP*, *BadDevice*, *BadMatch*, *BadAlloc*, and *BadValue* errors.

The *XGetDeviceModifierMapping* request returns a pointer to a newly created *XModifierKeymap* structure that contains the keys being used as modifiers. The structure should be freed after use by calling *XFreeModifierMapping*. If only zero values appear in the set for any modifier, that modifier is disabled.

*XGetDeviceModifierMapping* can generate *BadDevice* and *BadMatch* errors.

**The *XModifierKeymap* structure contains:**

```
typedef struct {  
    int max_keypermod;  
    KeyCode *modifiermap;  
} XModifierKeymap;
```

***BadLength*** More than eight keys were specified in the *XModifierKeymap* structure. ***BadAlloc*** The server failed to allocate the requested resource or server memory. ***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 *XGetDeviceModifierMapping* or *XChangeDeviceModifierMapping* 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.

**XSetDeviceKeyMapping(3X11)**

**XSetDeviceButtonMapping(3X11)**

*Programming With Xlib*