XSetDeviceButtonMapping, XGetDeviceButtonMapping – query or change device button mappings

int **XSetDeviceButtonMapping**(*display*, *device*, *map*, *nmap*)
   Display **display***;
   XDevice **device***;
   unsigned char *map*[]**;
   int *nmap***;

int **XGetDeviceButtonMapping**(*display*, *device*, *map_return*, *nmap*)
   Display **display***;
   XDevice **device***;
   unsigned char *map_return*[]**;
   int *nmap***;

display **Specifies the connection to the X server.** *device* **Specifies the device whose button mapping is to be queried or changed.** *map* **Specifies the mapping list.** *map_return* **Returns the mapping list.** *nmap* **Specifies the number of items in the mapping list.**

The *XSetDeviceButtonMapping* **request sets the mapping of the specified device. If it succeeds, the X server generates a** *DeviceMappingNotify* **event, and** *XSetDeviceButtonMapping* **returns** *MappingSuccess***. Element map[i] defines the logical button number for the physical button i+1. The length of the list must be the same as** *XGetDeviceButtonMapping* **would return, or a** *BadValue* **error results. A zero element disables a button, and elements are not restricted in value by the number of physical buttons. However, no two elements can have the same nonzero value, or a** *BadValue* **error results. If any of the buttons to be altered are logically in the down state,** *XSetDeviceButtonMapping* **returns** *MappingBusy***, and the mapping is not changed.**

XSetDeviceButtonMapping can generate *BadDevice*, *BadMatch*, and *BadValue* errors.

The *XGetDeviceButtonMapping* request returns the current mapping of the specified device. Buttons are numbered starting from one. *XGetDeviceButtonMapping* returns the number of physical buttons actually on the device. The nominal mapping for a device is map[i]=i+1. The nmap argument specifies the length of the array where the device mapping is returned, and only the first nmap elements are returned in map_return.

XGetDeviceButtonMapping can generate *BadDevice* or *BadMatch* errors.

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** *XGetDeviceButtonMapping* **or** *XSetDeviceButtonMapping* **request was made specifying a device that has no buttons.** *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.**

XChangeDeviceKeyboardControl(3X),
XChangeDeviceKeyMapping(3X)
XChangeDeviceModifierMapping(3X)
Programming With Xlib