

XSetPointerMapping, XGetPointerMapping – manipulate pointer settings

int XSetPointerMapping(*display*, *map*, *nmap*)

Display **display*;
unsigned char *map*[];
int *nmap*;

int XGetPointerMapping(*display*, *map_return*, *nmap*)

Display **display*;
unsigned char *map_return*[];
int *nmap*;

display Specifies the connection to the X server.
map Specifies the mapping list.
map_return Returns the mapping list.
nmap Specifies the number of items in the mapping list.

The **XSetPointerMapping** function sets the mapping of the pointer. If it succeeds, the X server generates a **MappingNotify** event, and **XSetPointerMapping** 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 **XGetPointerMapping** 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, **XSetPointerMapping** returns **MappingBusy**, and the mapping is not changed.

XSetPointerMapping can generate a **BadValue** error.

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

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.

XChangeKeyboardControl(3X11), XChangeKeyboardMapping(3X11)

Xlib – C Language X Interface