NAME

XkbSetNames - Change the symbolic names in the server

SYNOPSIS

Bool XkbSetNames (Display ****dpy*, unsigned int** *which***, unsigned int** *first_type***, unsigned int** *num_types***, XkbDescPtr** *xkb***);**

ARGUMENTS

dpy connection to the X server

which
mask of names or map components to be changed

first_type
first type whose name is to be changed

num_types
number of types for which names are to be changed

xkb keyboard description from which names are to be taken

DESCRIPTION

To change the symbolic names in the server, first modify a local copy of the keyboard description and then use either *XkbSetNames*, or, to save network traffic, use a XkbNameChangesRecstructure and call *XkbChangeNames* to download the changes to the server. *XkbSetNames* and *XkbChangeNames* can generate BadAlloc, BadAtom, BadLength, BadMatch, and BadImplementation errors.

Use *XkbSetNames* to change many names at the same time. For each bit set in *which, XkbSetNames* takes the corresponding value (or values in the case of arrays) from the keyboard description *xkb* and sends it to the server.

The *first_type* and *num_types* arguments are used only if XkbKeyTypeNamesMask or XkbKTLevelNamesMask is set in *which* and specify a subset of the types for which the corresponding names are to be changed. If either or both of these mask bits are set but the specified types are illegal, *XkbSetNames* returns False and does not update any of the names specified in *which.* The specified types are illegal if *xkb* does not include a map component or if *first_type* and *num_types* specify types that are not defined in the keyboard description.

STRUCTURES

The XkbNameChangesRec allows applications to identify small modifications to the symbolic names

and effectively reduces the amount of traffic sent to the server:

```
typedef struct _XkbNameChanges {
    unsigned int   changed;         /* name components that have changed */
    unsigned char  first_type;      /* first key type with a new name */
    unsigned char  num_types;       /* number of types with new names */
    unsigned char  first_lvl;       /* first key type with new level names */
    unsigned char  num_lvls;        /* number of key types with new level names */
    unsigned char  num_aliases;     /* if key aliases changed, total number of key aliases */
    unsigned char  num_rg;          /* if radio groups changed, total number of radio groups */
    unsigned char  first_key;       /* first key with a new name */
    unsigned char  num_keys;        /* number of keys with new names */
    unsigned short changed_vmods;   /* mask of virtual modifiers for which names have changed */
    unsigned long  changed_indicators; /* mask of indicators for which names were changed */
    unsigned char  changed_groups;  /* mask of groups for which names were changed */
} XkbNameChangesRec, *XkbNameChangesPtr
```

The *changed* field specifies the name components that have changed and is the bitwise inclusive OR of the valid names mask bits defined in Table 1. The rest of the fields in the structure specify the ranges that have changed for the various kinds of symbolic names, as shown in Table 2.

Xkb provides several functions that work with symbolic names. Each of these functions uses a mask to specify individual fields of the structures described above. These masks and their relationships to the fields in a keyboard description are shown in Table 1.

Table 1 Symbolic Names Masks

Mask Bit	Value	Keyboard Component	Field
XkbKeycodesNameMask	(1<<0)	Xkb->nameskeycodes	
XkbGeometryNameMask	(1<<1)	Xkb->namesgeometry	
XkbSymbolsNameMask	(1<<2)	Xkb->namessymbols	
XkbPhysSymbolsNameMask	(1<<3)	Xkb->namesphys_symbols	
XkbTypesNameMask	(1<<4)	Xkb->namestype	
XkbCompatNameMask	(1<<5)	Xkb->namescompat	
XkbKeyTypeNamesMask	(1<<6)	Xkb->map	type[*].name

XkbKTLevelNamesMask	(1<<7)	Xkb->map	type[*].lvl_names[*]
XkbIndicatorNamesMask	(1<<8)	Xkb->names	indicators[*]
XkbKeyNamesMask	(1<<9)	Xkb->names	keys[*], num_keys
XkbKeyAliasesMask	(1<<10)	Xkb->names	key_aliases[*], num_key_aliases
XkbVirtualModNamesMask	(1<<11)	Xkb->names	vmods[*]
XkbGroupNamesMask	(1<<12)	Xkb->names	groups[*]
XkbRGNamesMask	(1<<13)	Xkb->names	radio_groups[*], num_rg
XkbComponentNamesMask	(0x3f)	Xkb->names	keycodes, geometry, symbols, physical symbols, types, and compatibility map
XkbAllNamesMask	(0x3fff)	Xkb->names	all name components

Table 2 XkbNameChanges

Fields

Mask	Fields	Component	Field
XkbKeyTypeNamesMask	first_type, num_types	Xkb->map	type[*].name
XkbKTLevelNamesMask	first_lvl, num_lvls	Xkb->map	type[*].lvl_names[*]
XkbKeyAliasesMask	num_aliases	Xkb->names	key_aliases[*]
XkbRGNamesMask	num_rg	Xkb->names	radio_groups[*]
XkbKeyNamesMask	first_key, num_keys	Xkb->names	keys[*]
XkbVirtualModNamesMask	changed_vmods	Xkb->names	vmods[*]
XkbIndicatorNamesMask	changed_indicators	Xkb->names	indicators[*]
XkbGroupNamesMask	changed_groups	Xkb->names	groups[*]

DIAGNOSTICS

BadAlloc Unable to allocate storage

BadAtom A name is neither a valid Atom or None

BadImplementation
Invalid reply from server

BadLength The length of a request is shorter or longer than that required to minimally contain the arguments

BadMatch A compatible version of Xkb was not available in the server or an argument has correct type and range, but is otherwise invalid

SEE ALSO

XkbChangeNames(3)