NAME

XkbCopyKeyType - Copy one XkbKeyTypeRec structures

SYNOPSIS

Status XkbCopyKeyType (XkbKeyTypePtr *from***, XkbKeyTypePtr** *into***);**

ARGUMENTS

from

pointer to XkbKeyTypeRec to be copied

into

pointer to XkbKeyTypeRec to be changed

DESCRIPTION

XkbCopyKeyType copies the key type specified by *from* to the key type specified by *into.* Both must point to legal XkbKeyTypeRec structures. Xkb assumes *from* and *into* point to different places. As a result, overlaps can be fatal. *XkbCopyKeyType* frees any existing *map, preserve,* and *level_names* in *into* prior to copying. If any allocation errors occur while copying *from* to *into, XkbCopyKeyType* returns BadAlloc. Otherwise, *XkbCopyKeyType* copies *from* to *into* and returns Success.

STRUCTURES

Key types are used to determine the shift level of a key given the current state of the keyboard. The set of all possible key types for the Xkb keyboard description are held in the *types* field of the client map, whose total size is stored in *size_types,* and whose total number of valid entries is stored in *num_types.* Key types are defined using the following structure:

```
typedef struct {                /* Key Type */
   XkbModsRec       mods;       /* modifiers used to compute shift level */
   unsigned char    num_levels; /* total # shift levels, do not modify directly */
   unsigned char    map_count;  /* # entries in map, preserve (if non-NULL) */
   XkbKTMapEntryPtr map;        /* vector of modifiers for each shift level */
   XkbModsPtr       preserve;   /* mods to preserve for corresponding map entry */
   Atom             name;       /* name of key type */
   Atom *           level_names; /* array of names of each shift level */
} XkbKeyTypeRec, *XkbKeyTypePtr;
```

DIAGNOSTICS

BadAlloc Unable to allocate storage