NAME

XkbAllocCompatMap - Allocate a new compatibility map if you do not already have one available

SYNOPSIS

Status XkbAllocCompatMap (XkbDescPtr *xkb***, unsigned int** *which***, unsigned int** *num_si***);**

ARGUMENTS

xkb keyboard description in which to allocate compat map

which
mask of compatibility map components to allocate

num_si
number of symbol interpretations to allocate

DESCRIPTION

xkb specifies the keyboard description for which compatibility maps are to be allocated. The compatibility map is the *compat* field in this structure.

which specifies the compatibility map components to be allocated (see XkbGetCompatMap). *which* is an inclusive OR of the bits shown in Table 1.

Table 1 Compatibility Map Component Masks

Mask	Value	Affecting
XkbSymInterpMask	(1<<0)	Symbol interpretations
XkbGroupCompatMask	(1<<1)	Group maps
XkbAllCompatMask	(0x3)	All compatibility map components

num_si specifies the total number of entries to allocate in the symbol interpretation vector *(xkb.compat.sym_interpret).*

XkbAllocCompatMap returns Success if successful, BadMatch if *xkb* is NULL, or BadAlloc if errors are encountered when attempting to allocate storage.

STRUCTURES

```
typedef struct {
   KeySym          sym;         /* keysym of interest or NULL */
   unsigned char    flags;       /* XkbSI_AutoRepeat, XkbSI_LockingKey */
   unsigned char    match;       /* specifies how mods is interpreted */
   unsigned char    mods;        /* modifier bits, correspond to eight real modifiers */
   unsigned char    virtual_mod;  /* 1 modifier to add to key virtual mod map */
   XkbAnyAction     act;         /* action to bind to symbol position on key */
} XkbSymInterpretRec,*XkbSymInterpretPtr;
```

DIAGNOSTICS

BadAlloc Unable to allocate storage

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

XkbGetCompatMap(3)

NOTES

Note that symbol interpretations in a compatibility map (the *sym_interpret* vector of XkbSymInterpretRec structures) are also allocated using this same function. To ensure that there is sufficient space in the symbol interpretation vector for entries to be added, use *XkbAllocCompatMap* specifying *which* as XkbSymInterpretMask and the number of free symbol interpretations needed in *num_si*.