NAME

XkbAllocClientMap − Allocate and initialize an empty client map description record

SYNOPSIS

Status XkbAllocClientMap (XkbDescPtr *xkb***, unsigned int** *which***, unsigned int** *type_count***);**

ARGUMENTS

xkb keyboard description in which to allocate client map

which mask selecting map components to allocate

type_count
value of num_types field in map to be allocated

DESCRIPTION

Calling *XkbGetMap* should be sufficient for most applications to get client and server maps. As a result, most applications do not need to directly allocate client and server maps.

If you change the number of key types or construct map components without loading the necessary components from the X server, do not allocate any map components directly using *malloc* or *Xmalloc.* Instead, use the Xkb allocators, *XkbAllocClientMap,* and *XkbAllocServerMap.*

Similarly, use the Xkb destructors, *XkbFreeClientMap,* and *XkbFreeServerMap* instead of *free* or *Xfree.*

XkbAllocClientMap allocates and initializes an empty client map in the *map* field of the keyboard description specified by *xkb.* The *which* parameter specifies the particular components of the client map structure to allocate and is a mask composed by a bitwise inclusive OR of one or more of the masks shown in Table 1.

Table 1 XkbAllocClientMap Masks

Mask	Effect
XkbKeyTypesMask	The type_count field specifies the number of entries to preallocate for the types field of the client map. If the type_count field is less than XkbNumRequiredTypes returns BadValue.
XkbKeySymsMask	The min_key_code and max_key_code fields of the xkb parameter are used to allocate the syms and key_sym_map fields of the client map. The fields are allocated to contain the maximum number of entries necessary for max_key_code - min_key_code + 1 keys.
XkbModifierMapMask	The min_key_code and max_key_code fields of the xkb parameter are used to allocate the modmap field of the client map. The field is allocated to contain the maximum number of entries necessary for max_key_code - min_key_code + 1 keys.

NOTE: The *min_key_code* and *max_key_code* fields of the *xkb* parameter must be legal values if the XkbKeySymsMask or XkbModifierMapMask masks are set in the *which* parameter. If they are not valid, *XkbAllocClientMap* returns BadValue.

If the client map of the keyboard description is not NULL, and any fields are already allocated in the client map, *XkbAllocClientMap* does not overwrite the existing values; it simply ignores that part of the request. The only exception is the *types* array. If *type_count* is greater than the current *num_types* field of the client map, *XkbAllocClientMap* resizes the *types* array and resets the *num_types* field accordingly.

If *XkbAllocClientMap* is successful, it returns Success. Otherwise, it can return either BadMatch, BadAlloc, or BadValue errors.

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

BadValue An argument is out of range

SEE ALSO

XkbAllocClientMap(3), **XkbAllocServerMap**(3), **XkbFreeClientMap,**(3), **XkbFreeServerMap**(3), **XkbGetMap**(3)