

**NAME**

*XkbInitCanonicalKeyTypes* - Set the definitions of the canonical key types in a client map to their default values

**SYNOPSIS**

**Status** *XkbInitCanonicalKeyTypes* (**XkbDescPtr** *xkb*, **unsigned int** *which*, **int** *keypadVMod*);

**ARGUMENTS**

*xkb* keyboard description containing client map to initialize

*which*

mask of types to initialize

*keypadVMod*

index of NumLock virtual modifier

**DESCRIPTION**

*XkbInitCanonicalKeyTypes* initializes the first *XkbNumRequiredTypes* key types of the keyboard specified by the *xkb* parameter to their default values. The *which* parameter specifies what canonical key types to initialize and is a bitwise inclusive OR of the following masks: *XkbOneLevelMask*, *XkbTwoLevelMask*, *XkbAlphabeticMask*, and *XkbKeypadMask*. Only those canonical types specified by the *which* mask are initialized.

If *XkbKeypadMask* is set in the *which* parameter, *XkbInitCanonicalKeyTypes* looks up the NumLock named virtual modifier to determine which virtual modifier to use when initializing the KEYPAD key type. If the NumLock virtual modifier does not exist, *XkbInitCanonicalKeyTypes* creates it.

*XkbInitCanonicalKeyTypes* normally returns Success. It returns BadAccess if the Xkb extension has not been properly initialized, and BadAccess if the *xkb* parameter is not valid.

**RETURN VALUES**

Success            The *XkbInitCanonicalKeyTypes* function return Success if the Xkb extension has been properly initialized and the *xkb* parameter is valid.

**DIAGNOSTICS**

**BadAccess**        The Xkb extension has not been properly initialized