

**NAME**

XkbSASetPtrDfltValue - Sets the valueXXX field of act from val

**SYNOPSIS**

```
void XkbSASetPtrDfltValue (XkbAction act, int val);
```

**ARGUMENTS**

*act* action in which to set valueXXX

*val* value to set in valueXXX

**DESCRIPTION**

Actions associated with the XkbPtrDfltAction structure change the *mk\_dflt\_btn* attribute of the MouseKeys control.

If the MouseKeys control is not enabled, KeyPress and KeyRelease events are treated as though the action is XkbSA\_NoAction. Otherwise, this action changes the *mk\_dflt\_btn* attribute of the MouseKeys control.

The *type* field of the XkbPtrDfltAction structure should always be XkbSA\_SetPtrDflt.

The *flags* field is composed of the bitwise inclusive OR of the values shown in Table 1 (currently there is only one value defined).

Table 1 Pointer Default

Flags

-----  
Flag

Meaning

-----  
XkbSA\_DfltBtnAbsoluteIf set, the value field represents an absolute pointer button.

Otherwise, the value field represents the amount to be added to the current default button.

The *affect* field specifies what changes as a result of this action. The only valid value for the *affect* field is XkbSA\_AffectDfltBtn.

The *valueXXX* field is a signed character that represents the new button value for the *mk\_dflt\_btn* attribute of the MouseKeys control. If XkbSA\_DfltBtnAbsolute is set in *flags*, *valueXXX* specifies the button to be used; otherwise, *valueXXX* specifies the amount to be added to the current default button.

In either case, illegal button choices are wrapped back around into range. Xkb provides macros, to convert between the integer and signed character values in XkbPtrDfltAction structures.

## STRUCTURES

```
typedef struct _XkbPtrDfltAction {
    unsigned char  type;    /* XkbSA_SetPtrDflt */
    unsigned char  flags;  /* controls the pointer button number */
    unsigned char  affect; /* XkbSA_AffectDfltBtn */
    char          valueXXX; /* new default button member */
} XkbPtrDfltAction;
```