

NAME

XkbAddGeomSection - Add one section to an existing keyboard geometry

SYNOPSIS

```
XkbSectionPtr XkbAddGeomSection (XkbGeometryPtr geom, Atom name, int sz_rows, int  
sz_doodads, int sz_overlays);
```

ARGUMENTS

geom

geometry to be updated

name

name of the new section

sz_rows

number of rows to reserve in the section

sz_doodads

number of doodads to reserve in the section

sz_overlays

number of overlays to reserve in the section

DESCRIPTION

Xkb provides functions to add a single new element to the top-level keyboard geometry. In each case the *num_** fields of the corresponding structure is incremented by 1. These functions do not change *sz_** unless there is no more room in the array. Some of these functions fill in the values of the element's structure from the arguments. For other functions, you must explicitly write code to fill the structure's elements.

The top-level geometry description includes a list of *geometry properties*. A geometry property associates an arbitrary string with an equally arbitrary name. Programs that display images of keyboards can use geometry properties as hints, but they are not interpreted by Xkb. No other geometry structures refer to geometry properties.

A keyboard geometry contains an arbitrary number of sections. *XkbAddGeomSection* adds one section to an existing keyboard geometry *geom*. The new section contains space for the number of rows, doodads, and overlays specified by *sz_rows*, *sz_doodads*, and *sz_overlays*. The new section is allocated and zeroed and given the name specified by *name*. If a section with name *name* already exists in the geometry, a pointer to the existing section is returned. *XkbAddGeomSection* returns

NULL if any of the parameters is empty or if it was not able to allocate space for the section. To allocate space for an arbitrary number of sections to a geometry, use *XkbAllocGeomSections*.

STRUCTURES

```
typedef struct _XkbSection {
    Atom          name;      /* section name */
    unsigned char priority; /* drawing priority, 0=>highest, 255=>lowest */
    short         top;      /* top coordinate of section origin */
    short         left;     /* left coordinate of row origin */
    unsigned short width;   /* section width, in mm/10 */
    unsigned short height;  /* section height, in mm/10 */
    short         angle;    /* angle of section rotation, counterclockwise */
    unsigned short num_rows; /* number of rows in the rows array */
    unsigned short num_doodads; /* number of doodads in the doodads array */
    unsigned short num_overlays; /* number of overlays in the overlays array */
    unsigned short sz_rows; /* size of the rows array */
    unsigned short sz_doodads; /* size of the doodads array */
    unsigned short sz_overlays; /* size of the overlays array */
    XkbRowPtr     rows;     /* section rows array */
    XkbDoodadPtr  doodads;  /* section doodads array */
    XkbBoundsRec  bounds;   /* bounding box for the section, before rotation */
    XkbOverlayPtr overlays; /* section overlays array */
} XkbSectionRec, *XkbSectionPtr;
```

top and *left* are the origin of the section, relative to the origin of the keyboard, in mm/10. *angle* is in 1/10 degrees.

SEE ALSO

XkbAllocGeomSections(3)