

**NAME**

XkbAddGeomShape - Add a shape to a keyboard geometry

**SYNOPSIS**

```
XkbShapePtr XkbAddGeomShape (XkbGeometryPtr geom, Atom name, int sz_outlines);
```

**ARGUMENTS**

*geom*

geometry to be updated

*name*

name of the new shape

*sz\_outlines*

number of outlines to be reserved

**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 geometry contains an arbitrary number of shapes, each of which is made up of an arbitrary number of outlines. *XkbAddGeomShape* adds a shape to a geometry *geom* by allocating space for *sz\_outlines* outlines for it and giving it the name specified by *name*. If a shape with name *name* already exists in the geometry, a pointer to the existing shape is returned. *XkbAddGeomShape* returns NULL if any of the parameters is empty or if it was not able to allocate space. To allocate space for an arbitrary number of geometry shapes, use *XkbAllocGeomShapes*.

**STRUCTURES**

```
typedef struct _XkbShape {
    Atom      name;      /* shape's name */
    unsigned short  num_outlines; /* number of outlines for the shape */
}
```

```
    unsigned short  sz_outlines; /* size of the outlines array */
    XkbOutlinePtr   outlines;    /* array of outlines for the shape */
    XkbOutlinePtr   approx;     /* pointer into the array to the approximating outline */
    XkbOutlinePtr   primary;    /* pointer into the array to the primary outline */
    XkbBoundsRec    bounds;     /* bounding box for the shape; encompasses all outlines */
} XkbShapeRec, *XkbShapePtr;
```

**SEE ALSO****XkbAllocGeomShapes(3)**