

**NAME**

XDrawRectangle, XDrawRectangles, XRectangle - draw rectangles and rectangles structure

**SYNTAX**

```
int XDrawRectangle(Display *display, Drawable d, GC gc, int x, int y, unsigned int width, unsigned int height);
```

```
int XDrawRectangles(Display *display, Drawable d, GC gc, XRectangle rectangles[], int nrectangles);
```

**ARGUMENTS**

<i>d</i>	Specifies the drawable.
<i>display</i>	Specifies the connection to the X server.
<i>gc</i>	Specifies the GC.
<i>nrectangles</i>	Specifies the number of rectangles in the array.
<i>rectangles</i>	Specifies an array of rectangles.
<i>width</i>	
<i>height</i>	Specify the width and height, which specify the dimensions of the rectangle.
<i>x</i>	
<i>y</i>	Specify the x and y coordinates, which specify the upper-left corner of the rectangle.

**DESCRIPTION**

The **XDrawRectangle** and **XDrawRectangles** functions draw the outlines of the specified rectangle or rectangles as if a five-point **PolyLine** protocol request were specified for each rectangle:

```
[x,y] [x+width,y] [x+width,y+height] [x,y+height] [x,y]
```

For the specified rectangle or rectangles, these functions do not draw a pixel more than once.

**XDrawRectangles** draws the rectangles in the order listed in the array. If rectangles intersect, the intersecting pixels are drawn multiple times.

Both functions use these GC components: function, plane-mask, line-width, line-style, cap-style, join-style, fill-style, subwindow-mode, clip-x-origin, clip-y-origin, and clip-mask. They also use these GC

mode-dependent components: foreground, background, tile, stipple, tile-stipple-x-origin, tile-stipple-y-origin, dash-offset, and dash-list.

**XDrawRectangle** and **XDrawRectangles** can generate **BadDrawable**, **BadGC**, and **BadMatch** errors.

## STRUCTURES

The **XRectangle** structure contains:

```
typedef struct {  
    short x, y;  
    unsigned short width, height;  
} XRectangle;
```

All x and y members are signed integers. The width and height members are 16-bit unsigned integers. You should be careful not to generate coordinates and sizes out of the 16-bit ranges, because the protocol only has 16-bit fields for these values.

## DIAGNOSTICS

**BadDrawable** A value for a Drawable argument does not name a defined Window or Pixmap.

**BadGC** A value for a GCContext argument does not name a defined GCContext.

**BadMatch** An **InputOnly** window is used as a Drawable.

**BadMatch** Some argument or pair of arguments has the correct type and range but fails to match in some other way required by the request.

## SEE ALSO

XDrawArc(3), XDrawLine(3), XDrawPoint(3)

*Xlib - C Language X Interface*