

**NAME**

XDrawText, XDrawText16, XTextItem, XTextItem16 - draw polytext text and text drawing structures

**SYNTAX**

```
int XDrawText(Display *display, Drawable d, GC gc, int x, int y, XTextItem *items, int nitems);
```

```
int XDrawText16(Display *display, Drawable d, GC gc, int x, int y, XTextItem16 *items, int nitems);
```

**ARGUMENTS**

- |                |                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <i>d</i>       | Specifies the drawable.                                                                                                                   |
| <i>display</i> | Specifies the connection to the X server.                                                                                                 |
| <i>gc</i>      | Specifies the GC.                                                                                                                         |
| <i>items</i>   | Specifies an array of text items.                                                                                                         |
| <i>nitems</i>  | Specifies the number of text items in the array.                                                                                          |
| <i>x</i>       |                                                                                                                                           |
| <i>y</i>       | Specify the x and y coordinates, which are relative to the origin of the specified drawable and define the origin of the first character. |

**DESCRIPTION**

The **XDrawText16** function is similar to **XDrawText** except that it uses 2-byte or 16-bit characters. Both functions allow complex spacing and font shifts between counted strings.

Each text item is processed in turn. A font member other than **None** in an item causes the font to be stored in the GC and used for subsequent text. A text element delta specifies an additional change in the position along the x axis before the string is drawn. The delta is always added to the character origin and is not dependent on any characteristics of the font. Each character image, as defined by the font in the GC, is treated as an additional mask for a fill operation on the drawable. The drawable is modified only where the font character has a bit set to 1. If a text item generates a **BadFont** error, the previous text items may have been drawn.

For fonts defined with linear indexing rather than 2-byte matrix indexing, each **XChar2b** structure is interpreted as a 16-bit number with byte1 as the most significant byte.

Both functions use these GC components: function, plane-mask, fill-style, font, 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, and tile-stipple-y-origin.

**XDrawText** and **XDrawText16** can generate **BadDrawable**, **BadFont**, **BadGC**, and **BadMatch** errors.

## STRUCTURES

The **XTextItem** and **XTextItem16** structures contain:

```
typedef struct {
    char *chars; /* pointer to string */
    int nchars; /* number of characters */
    int delta; /* delta between strings */
    Font font; /* Font to print it in, None don't change */
} XTextItem;
```

```
typedef struct {
    XChar2b *chars; /* pointer to two-byte characters */
    int nchars; /* number of characters */
    int delta; /* delta between strings */
    Font font; /* font to print it in, None don't change */
} XTextItem16;
```

If the font member is not **None**, the font is changed before printing and also is stored in the GC. If an error was generated during text drawing, the previous items may have been drawn. The baseline of the characters are drawn starting at the x and y coordinates that you pass in the text drawing functions.

For example, consider the background rectangle drawn by **XDrawImageString**. If you want the upper-left corner of the background rectangle to be at pixel coordinate (x,y), pass the (x,y + ascent) as the baseline origin coordinates to the text functions. The ascent is the font ascent, as given in the **XFontStruct** structure. If you want the lower-left corner of the background rectangle to be at pixel coordinate (x,y), pass the (x,y - descent + 1) as the baseline origin coordinates to the text functions. The descent is the font descent, as given in the **XFontStruct** structure.

## DIAGNOSTICS

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

**BadFont** A value for a Font or GCcontext argument does not name a defined Font.

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

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

**SEE ALSO**

XDrawImageString(3), XDrawString(3), XLoadFont(3)

*Xlib - C Language X Interface*