**drawText( text, location [, fgColor] [, flags] [, tagID])**
The drawText() function is a simplified text drawing function that was introduced in EFS2. It accepts no barIndex parameter so it can only be used to draw text at the current bar offset.

Parameters

|  |  |
| --- | --- |
| text  | the symbol or text string to draw  |
| location  | see Location Flags below  |
| fgColor  | optional. the foreground [color](http://kb.esignal.com/article.aspx?article=2148&p=4) to use  |
| flags  | optional. see Text Flags below  |
| tagID  | optional. a unique identifier for this text object or null  |

Location Flags - **Text.PRESET MUST be used if you use one of these for the yValue…NO other vertical flags can be used!!!**

|  |  |
| --- | --- |
| TopRow1  | draws at the upper margin of the chart  |
| TopRow2  | draws one row down from TopRow1  |
| TopRow3  | draws one row down from TopRow2  |
| TopRow4  | draws one row down from TopRow3  |
| AboveBar1  | draws directly above the high of the bar  |
| AboveBar2  | draws one row above AboveBar1  |
| AboveBar3  | draws one row above AboveBar2  |
| AboveBar4  | draws one row above AboveBar3  |
| BelowBar1  | draws directly below the low of the bar  |
| BelowBar2  | draws one row below BelowBar1  |
| BelowBar3  | draws one row below BelowBar2  |
| BelowBar4  | draws one row below BelowBar3  |
| BottomRow1  | draws at the lower margin of the chart  |
| BottomRow2  | draws one row above BottomRow1  |
| BottomRow3  | draws one row above BottomRow2  |
| BottomRow4  | draws one row above BottomRow3  |

Text Flags

|  |  |
| --- | --- |
| Text.BOLD  | display the text in bold  |
| Text.ITALIC  | display the text in italics  |
| Text.UNDERLINE  | underline the text  |
| Text.PLAIN  | display plain text  |
| Text.LEFT | default - Left align text with x/y value |
| Text.RIGHT | right align text with x/y value |
| Text.TOP | default - top align text with x/y value |
| Text.BOTTOM | bottom align text with x/y value |
| Text.ONTOP | Draws the text on top of the study, otherwise the study will be drawn on top of the text |
| Text.FRAME | draws a frame around the text using FGColor as the frame color |
| Text.BUTTON | draws the text as a button. FGColor and BGColor are ignored |
| Text.RELATIVETOLEFT | keeps text fixed at a relative bar distance from left chart axis |
| Text.RELATIVETOBOTTOM | keeps text fixed at a relative pixel distance from bottom chart axis |
| Text.RELATIVETOTOP | keeps text fixed at a relative pixel distance from the top chart axis |
| Text.CENTER | horizontally centers the text over the bar |
| Text.VCENTER | vertically center text at the price value |
| **Text.PRESET** | **MUST be used if you use the location flags as the yValue…you can’t use any other y Flags!** |
| Note: Text flags can be ORed together to combine various attributes (*e.g*., Text.ITALIC | Text.BOLD | Text.UNDERLINE)  |

Usage – AboveBar1, 2, 3 stacks 3 different text items above the current bar.

drawTextRelative(0, AboveBar1, "\u00EA", Color.red, null, Text.PRESET|Text.CENTER, "Wingdings", 10, "Exit"+rawtime(0));

drawTextRelative(0, AboveBar2, "\u00CC", Color.red, null, Text.PRESET|Text.CENTER, "Wingdings", 10, "Exit2"+rawtime(0));

drawTextRelative(0, AboveBar3, "Sell", Color.red, null, Text.PRESET|Text.CENTER, "Arial", 10, "Exit3"+rawtime(0));

drawText("Suggested Long Exit",BottomRow1,Color.red,Text.LEFT,"TextExit"+rawtime(0));

**Go to Start -> Windows Accessories -> Character Map -> Font: Wingdings** ---- From the dialog box you can see the various characters to use.

**Translate the Hex character code of the Wingding font into a Unicode escape sequence** which is easily done, as follows:

* Pull up character map and click on the particular symbol that you want to print on an eSignal chart.
* Look at the bottom of the character map application and you will see the Character code. In your example of the right arrow, the character code is 0x84
* The Unicode escape sequence for **0x84 is simply '\u0084'**. The Unicode sequence will always be '\u00' (those are two zeros) concatenated to the last 2 characters displayed in the Character Map character code.
* Put that in your drawTextRelative function and it will print the correct symbol.

**drawTextRelative**(xBar, yValue, Text, [FGColor], [BGColor], [Flags], [FontName], [FontSize], [TagName], [cx], [cy])

This function is used to draw text on the chart. Unlike the drawTextAbsolute() function, the text will 'stick' to the x-axis position that it was originally written to. As new bars come in, the text will shift to the left along with the bars. The TagName parameter must be unique for each text object that is drawn on the chart. If, for example, you call the drawText() functions repeatedly in your script but you always use a TagName of "1", then only one text object will appear on your chart (the last one drawn).

* *xBar*:  Relative position where text should appear.
* *yValue*:  y-axis value where text should appear
* *Text:*The text that should be displayed
* *FGColor*:  Optional. If not provided, pass null. Foreground color of the text.
* *BGColor*:  Optional. if not provided, pass null. Background color of the text
* *Flags*:  Text Flags (these can be ORd together). Pass null if not using flags.
* *FontName*:  Optional. If not provided, pass null. Otherwise, pass a font name (e.g, "Courier" or "Arial").
* *FontSize*:  Optional. If not provided, pass null. Otherwise, pass the font size to use (e.g., 11 or 15 or 8, etc.)
* *TagName*:  A unique identifier for this text object.
* *cx*:   Optional. Pixel spacing control.
* *cy*:   Optional. Pixel spacing control.

**Note re cx/cy**:

The cx and cy parameters control the width (cx) and height (cy) of the text label. They are not very useful unless you are using them in combination with text flags, RELATIVETOTOP, RELATIVETOLEFT and RELATIVETOBOTTOM.  Without these flags the parameters will be relative to the bar index for cx and the price scale for cy.
Both cx and cy require whole numbers.

You can pass positive or negative numbers to these parameters.  If you use positive whole numbers then the size is based on that number of pixels.

* cx of 15 will be the width of the text label of 15 pixels.
* cy of 15 will be the height of the text label of 15 pixels.

If you use negative whole numbers then the size is relative to the specified font size.

* cx of -15 will be the approximate width of 15 characters of the specified font.
* cy of -2 will be 2 times the height of the specified font. -3 would be 3 times the height etc.
* cy == 0 uses size of text.