Parameters

fontNameStr	A string expression containing the name of a font to be used for generating outlines.
fstyle	*fstyle* is a bitwise parameter with each bit controlling one aspect of the font style as follows:: Bit 0: Bold Bit 1: Italic Bit 2: Underline Bit 4: Strikethrough See **Setting Bit Parameters** on page IV-12 for details about bit settings.
textStr	A string expression containing the text to be transformed into Bezier outlines.
xWaveName	Specifies X output wave to receive the Bezier curve data.
yWaveName	Specifies Y output wave to receive the Bezier curve data.

Flags

/FS=*fs*	*fs* is the font size to apply while generating the outlines. Without this flag, glyphs are scaled to unit size. The scaling parameters for DrawBezier are limited to a maximum of 20, so you will probably need to use this flag if you want very large text.
/O	Allow overwriting the output waves.

Details

To draw the text outline, pass the output waves from Text2Bezier to the DrawBezier operation. The coordinates are such that the DrawBezier origin will be the left baseline of the text. In most cases you will want to use the SetDrawEnv operation to set the coordinate system to absolute. If you wish to fill the text with color or gradient, you will need to use the SetDrawEnv subpaths keyword.

Output Variables

V_Flag	0 for success, 1 for general failure, 2 for extraction failure.

Examples

```
// Extract text to Bezier data
Text2Bezier/O/FS=20 "Helvetica", 0, "Text2Bezier", textx, texty

// Draw outline in a graph window with no fill
Display /W=(100,100,700,450)
SetDrawLayer UserFront
SetDrawEnv fillpat=0,xcoord= abs,ycoord= abs// fillpat=0 specifies no fill
DrawBezier 100,100,1,1,textx,texty

// Draw outline three times larger and filled with a gradient shading
// subpaths=1 draws the entire Bezier as a series of subpaths for correct filling
SetDrawEnv xcoord= abs,ycoord= abs,subpaths= 1,
                      gradient= {0, 0, 0, 1, 0, (65535,65535,0), (65535,0,0)}
DrawBezier 100,150,3,3,textx,texty
```

See Also

DrawBezier, **SetDrawEnv**

TextBox

TextBox [***flags***] [***textStr***]

The TextBox operation puts a textbox on the target or named graph window. A textbox is an annotation that is not associated with any particular trace.

Parameters

textStr is the text that is to appear in the textbox. It is optional.

Flags

/A=*anchorCode* Specifies position of textbox anchor point.

anchorCode	Position	*anchorCode*	Position
LT	left top	RT	right top
LC	left center	RC	right center
LB	left bottom	RB	right bottom
MT	middle top		
MC	middle center		
MB	middle bottom		

anchorCode is a literal, *not* a string.

For interior textboxes, the anchor point is on the rectangular edge of the plot area of the graph window (where the left, bottom, right, and top axes are drawn).

For exterior textboxes, the anchor point is on the rectangular edge of the entire graph window.

/B=(*r*,*g*,*b*[,*a*]) Sets the color of the textbox background. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**.

/B=*b* Controls the textbox background.

b=0: Opaque background.

b=1: Transparent background.

b=2: Same background as the graph plot area background.

b=3: Same background as the window background.

/C Changes existing textbox.

/D={*thickMult* [, *shadowThick* [, *haloThick*]]}

thickMult multiplies the normal frame thickness of a text-box. The thickness may be set using just /D=*thickMult*.

shadowThick, if present, overrides Igor's normal shadow thickness. It is in units of fractional points.

haloThick governs the annotation's halo thickness (a surrounding band of the annotation's background color), which can be -1 to 10 points wide.

The default *haloThick* value is -1, which preserves the behavior of previous versions of Igor where the halo of all annotations was set by the global variable root:V_TBBufZone. Any negative value of *haloThick* (-0.5, for example) will be overridden by V_TBBufZone if it exists, otherwise the absolute value of *haloThick* will be used. A zero or positive value overrides V_TBBufZone.

Any of the parameters may be missing. To set *haloThick* to 0 without changing other parameters, use `/D={,,0}`.

/E[=*exterior*] /E or /E=1 forces textbox (or legend) to be exterior to graph (provided *anchorCode* is not MC) and pushes the graph margins away from the anchor edge(s). /E=2 also forces exterior mode but does not push the margins.

/E=0 returns it to the default (an "interior textbox" which can be anywhere in the graph window).

/F=*frame* Controls the textbox frame.

frame=0: No frame.

frame=1: Underline frame.

frame=2: Box frame.

/G=(*r*,*g*,*b*[,*a*]) Sets color of the text in the textbox. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**.

/H=*legendSymbolWidth*

legendSymbolWidth sets width of the legend symbol (the sample line or marker) in points. Use 0 for the default, automatic width.

/K Kills existing textbox.

/LS= *linespace* Specifies a tweak to the normal line spacing where *linespace* is points of extra (plus or minus) line spacing. For negative values, a blank line may be necessary to avoid clipping the bottom of the last line.

/M[=*sameSize*] /M or /M=1 specifies that legend markers should be the same size as the marker in the graph.

/M=0 turns same-size mode off so that the size of the marker in the legend is based on text size.

/N=*name* Specifies the name of the textbox to change or create.

/O=*rot* Sets the text's rotation. *rot* is in (integer) degrees, counterclockwise and must be a number from -360 to 360.

0 is normal horizontal left-to-right text, 90 is vertical bottom-to-top text.

/R=*newName* Renames the textbox.

/S=*style* Controls the textbox frame style.

style=0: Single frame.

style=1: Double frame.

style=2: Triple frame.

style=3: Shadow frame.

/T=*tabSpec* *tabSpec* is a single number in points, such as `/T=72`, for evenly spaced tabs or a list of tab stops in points such as `/T={50, 150, 225}`.

/V=*vis* Controls annotation visibility.

vis=0: Invisible annotation; not selectable. The annotation is still listed in **AnnotationList**.

vis=1: Visible annotation (default).

/W=*winName* Operates in the named graph window or subwindow. When omitted, action will affect the active window or subwindow. This must be the first flag specified when used in a Proc or Macro or on the command line.

When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

/X=*xOffset* For interior textboxes *xOffset* is the distance from anchor to textbox as a percentage of the plot area width.

For exterior textboxes *xOffset* is the distance from anchor to textbox as a percentage of the graph window width. See /E and /A.

/Y=*yOffset* *yOffset* is the distance from anchor to textbox as a percentage of the plot area height (interior textboxes) or graph window height (exterior textboxes). See /E and /A.