/Z=*freeze* Controls freezing of textbox position.

freeze=1: Freezes textbox position (you can't move it with the mouse).

freeze=0: Unfreezes it.

Details

Use the optional /W=*winName* flag to specify a specific graph or layout window. When used on the command line or in a Macro, Proc, or Window procedure, /W must precede all other flags.

If the /C flag is used, it must be the first flag in the command (except that if may follow an initial /W) and must be followed immediately by the /N=*name* flag.

If the /K flag is used, it must be the first flag in the command (or follow an initial /W) and must be followed immediately by the /N=*name* flag with no further flags or parameters.

textStr is optional. If missing, the textbox text is unchanged. This allows changes to the textbox to be made through the flags without changing the text.

A textbox can have at most 100 lines.

textStr can contain escape codes which affect subsequent characters in the text. An escape code is introduced by a backslash character. In a literal string, you must enter two backslashes to produce one. See **Backslashes in Annotation Escape Sequences** on page III-58 for details.

Using escape codes you can change the font, size, style and color of text, create superscripts and subscripts, create dynamically-updated text, insert legend symbols, and apply other effects. See **Annotation Escape Codes** on page III-53 for details.

The characters "<??>" in a textbox indicate that you specified an invalid escape code or used a font that is not available.

Examples

```
TextBox/C/N=t1/X=25/Y=50
```

moves the textbox named t1 to the location defined by X=25 and Y=50.

```
TextBox/C/N=t1 "New Text"
```

changes the text for t1.

See Also

Tag, **Legend**, **AppendText**, **AnnotationInfo**, **AnnotationList**

Annotation Escape Codes on page III-53

See the **printf** operation for formatting codes used in *formatStr*.

Programming with Annotations on page III-52.

Trace Names on page II-282, **Programming With Trace Names** on page IV-87.

TextEncoding

```
#pragma TextEncoding = "<text encoding name>"
```

#pragma TextEncoding is a compiler directive that tells Igor the text encoding used by a procedure file. Igor needs to know this in order to correctly interpret non-ASCII characters in the file. We recommend that you add a TextEncoding pragma to your procedure files.

All new procedure files should use the UTF-8 text encoding:

```
#pragma TextEncoding = "UTF-8"
```

See **Text Encoding Names and Codes** on page III-490 for a list of accepted text encoding names.

This statement must be flush against the left edge of the procedure file with no indentation. It is usually placed at or near the top of the file.

The TextEncoding pragma was added in Igor Pro 7.00 and is ignored by earlier versions.

See **The TextEncoding Pragma** on page IV-55 for further explanation.