reverse is optional. If missing, it is taken to be 0. If *reverse* is nonzero then the sense of the match is reversed. For example, if *regExprStr* is `"^abc"` and *reverse* is `1`, then all list items that do not start with "abc" are returned.

listSepStr is optional; the default is `";"`. In order to specify *listSepStr*, you must precede it with reverse.

Examples

To list ColorTables containing "Red", "red", or "RED" (etc.):

```
Print GrepList(CTabList(),"(?i)red")        // case-insensitive matching
```

To list window recreation commands starting with "\tCursor":

```
Print GrepList(WinRecreation("Graph0", 0), "^\tCursor", 0 , "\r")
```

See Also

Regular Expressions on page IV-176.

ListMatch, **StringFromList**, and **WhichListItem** functions and the **Grep** operation.

GrepString

`GrepString(`*`string`*`, `*`regExprStr`*`)`

The GrepString function tests *string* for a match to the regular expression *regExprStr*. Returns 1 to indicate a match, or 0 for no match.

Details

regExprStr is a regular expression such as is used by the UNIX grep(1) command. It is much more powerful than the wildcard syntax used for **StringMatch**. See **Regular Expressions** on page IV-176 for *regExprStr* details.

Character matching is case-sensitive by default, similar to **strsearch**. Prepend the Perl 5 modifier "(?i)" to match upper and lower-case text

Examples

Test for truth that the string contains at least one digit:

```
if( GrepString(str,"[0-9]+") )
```

Test for truth that the string contains at least one "abc", "Abc", "ABC", etc.:

```
if( GrepString(str,"(?i)abc") )             // case-insensitive test
```

See Also

Regular Expressions on page IV-176.

The **StringMatch**, **CmpStr**, **strsearch**, **ListMatch**, and **ReplaceString** functions and the **Demo** and **sscanf** operations.

GridStyle

`GridStyle`

GridStyle is a procedure subtype keyword that puts the name of the procedure in the Grid->Style Function submenu of the mover pop-up menu in the drawing tool palette. You can have Igor automatically create a grid style function for you by choosing Save Style Function from that submenu.

GroupBox

`GroupBox` `[`**`/Z`**`]` ***`ctrlName`*** `[`***`keyword = value`*** `[,` ***`keyword = value`*** `…]]`

The GroupBox operation creates a box to surround and group related controls.

For information about the state or status of the control, use the **ControlInfo** operation.

Parameters

ctrlName is the name of the GroupBox control to be created or changed.

The following keyword=value parameters are supported:

align=*alignment*

Sets the alignment mode of the control. The alignment mode controls the interpretation of the *leftOrRight* parameter to the pos keyword. The align keyword was added in Igor Pro 8.00.

If *alignment*=0 (default), *leftOrRight* specifies the position of the left end of the control and the left end position remains fixed if the control size is changed.

If *alignment*=1, *leftOrRight* specifies the position of the right end of the control and the right end position remains fixed if the control size is changed.

appearance={*kind* [, *platform*]}

Sets the appearance of the control. *platform* is optional. Both parameters are names, not strings.

kind can be one of `default`, `native`, or `os9`.

platform can be one of `Mac`, `Win`, or `All`.

See **DefaultGUIControls Default Fonts and Sizes** for how enclosed controls are affected by native groupbox appearance.

See **Button** for more appearance details.

disable=*d*

Sets user editability of the control.

d=0:	Normal.
d=1:	Hide.
d=2:	Draw in gray state.

fColor=(*r*,*g*,*b*[,*a*])

Sets color of the title text. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**.

font="*fontName*"

Sets font used for the box title, e.g., `font="Helvetica"`.

frame=*f*

Sets frame mode. If 1 (default), the frame has a 3D look. If 0, then a simple gray line is used. Generally, you should not use frame=0 with a title if you want to be in accordance with human interface guidelines.

fsize=*s*

Sets font size for box title.

fstyle=fs

Sets the font style of the title text. *fs* is a bitwise parameter with each bit controlling one aspect of the font style for the tick mark labels 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.

help={*helpStr*}

Sets the help for the control.

helpStr is limited to 1970 bytes (255 in Igor Pro 8 and before).

You can insert a line break by putting “\r” in a quoted string.

labelBack=(*r*,*g*,*b*[,*a*]) or 0

Sets fill color for the interior. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**.

If you do not set labelBack then the interior is transparent.

If an opaque fill color is used, drawing objects can not be used because they will be covered up. Also, you will have to make sure the GroupBox is drawn before any interior controls.

The fidelity of the coloring is platform-dependent.