

Keyword	Information Following Keyword
FONT	The name of the column's font.
SIZE	Column's font size.
STYLE	As specified for the ModifyTable operation's style keyword.
ALIGNMENT	0=left, 1=center, 2=right.
RGB	The column's color in R,G,B format.
RGBA	The column's color in R,G,B,A format. Added in Igor Pro 9.00.
ELEMENTS	As specified for the ModifyTable operation's elements keyword.

Examples

This example makes the table's target cell advance by one position within the range of selected cells each time it is called. To try it, create a table, select a range of cells and then run the function using the Macros menu.

```
Menu "Macros"
    "Test/1", /Q, AdvanceTargetCell("")
End

Function AdvanceTargetCell(tableName)
    String tableName           // Name of table or "" for top table.

    String info = TableInfo(tableName, -2)
    if (strlen(info) == 0)
        return -1             // No such table
    endif

    String selectionInfo
    selectionInfo = StringByKey("SELECTION", info)

    Variable fRow, fCol, lRow, lCol, tRow, tCol
    sscanf selectionInfo, "%d,%d,%d,%d,%d", fRow, fCol, lRow, lCol, tRow, tCol

    tCol += 1
    if (tCol > lCol)
        tCol = fCol
        tRow += 1
        if (tRow > lRow)
            tRow = fRow
        endif
    endif

    ModifyTable selection=(-1, -1, -1, -1, tRow, tCol)
End
```

See Also

The **ModifyTable** operation.

Tag

Tag [flags] [taggedObjectName, xAttach [, textStr]]

The Tag operation puts a tag on the target or named graph window or subwindow. A tag is an annotation that is attached to a particular point on a trace, image, waterfall plot, or axis in a graph.

The Tag operation can be invoked in several ways as illustrated by these examples:

```
// Make a wave and a graph
Make wave0 = sin(x/8); Display wave0

// Tag command with all optional parameters included
Tag/N=tag0 wave0, 0, "Point 0 on wave0"          // Add a tag to a trace

// Tag command with all optional parameters omitted
Tag/C/N=tag0/F=0                                  // Change frame using /F flag

// Tag command with optional text parameter omitted
Tag/C/N=tag0 wave0, 50                            // Change the tagged point
```

Tag

taggedObjectName and *xAttach* can be omitted only when changing an existing tag using Tag/C/N=<tag name>. *textStr* can be included only if *taggedObjectName* and *xAttach* are included.

Parameters

taggedObjectName is a trace, image or axis name which identifies the object to which the tag is to be attached. The name can be optionally followed by the # character and an instance number to distinguish multiple trace or image instances of the same wave. An axis name can be one of the standard axis names (bottom, top, left, or right) or a user-defined free axis name.

The *taggedObjectName* parameter is a name so, if you have a string variable containing the name, you must precede the string variable name with the \$ operator.

xAttach identifies the point on the trace, image, or axis, to which the tag is to be attached. See **xAttach Parameter** below for details.

textStr is the text that is to appear in the tag.

xAttach Parameter

xAttach identifies the point on the trace, image, or axis, to which the tag is to be attached.

For a trace tag, *xAttach* is the X wave scaling value of the attachment point.

For an image tag, *xAttach* is the X index in terms of the image wave's X scaling of the wave element to be tagged treating the wave as if it were 1D.

For a horizontal axis tag, *xAttach* is the X axis value of the point on the axis to which the tag is to be attached. Specifying NaN for *xAttach* centers the tag on the axis.

For a vertical axis tag, *xAttach* is the Y axis value of the point on the axis to which the tag is to be attached. Specifying NaN for *xAttach* centers the tag on the axis.

Flags

/A=*anchorCode* Specifies position of tag anchor point. *anchorCode* is a literal, *not* a string.

LT	left top
LC	left center
LB	left bottom
MT	middle top
MC	middle center (default)
MB	middle bottom
RT	right top
RC	right center
RB	right bottom

The anchor point is on the tag itself. Any line or arrow drawn from the tag to the wave starts at the tag's anchor point. The anchor point also determines the precise spot on the tag which represents its position.

/AO=*a0* Sets the text's auto-orientation mode. A non-zero *a0* value overrides the /O value.

/AO is for trace tags only. Setting /AO for any other kind of annotation has no effect.

An auto-oriented tag's text rotates whenever it is redrawn, usually when the underlying data changes, the graph is resized, or when the tag is attached to a new point.

The values for *ao* are:

- ao*=0: No auto-orientation. Use the /O value (default).
- ao*=1: Tangent to the trace line at the attachment point.
- ao*=2: Tangent to the trace line, snaps to vertical or horizontal if within 2 degrees of vertical or horizontal.
- ao*=3: Perpendicular to the trace line.
- ao*=4: Perpendicular to the trace line, snaps to vertical or horizontal if within 2 degrees of vertical or horizontal.

/AXS[=*isAxisTag*] Specifies that *taggedObjectName* is to be interpreted as an axis name. This is useful when the axis name is the same as one of the graph's trace or image instance names. /AXS is the same as /AXS=1. The /AXS flag was added in Igor Pro 9.00.

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

/B=*b* Controls the tag 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 the existing tag.

/F=*frame* Controls the tag 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 tag. *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.

/I=*i* Controls the tag visibility.

- i*=1: Tag will be invisible if it is "off screen". "Off screen" means that its attachment point or any part of the tag's text is off screen. This is esthetically pleasing but gives you nothing to grab if you want to drag the tag back on screen.
- i*=0: Tag will always be visible. If it is "off screen", it appears at the extreme edge of the graph.

/IMG[=*isAxisTag*] Specifies that *taggedObjectName* is to be interpreted as an image instance name. This is useful when the image instance name is the same as one of the graph's trace instance names or one of the axis names. /IMG is the same as /IMG=1. The /IMG flag was added in Igor Pro 9.00.

/K Kills existing tag.

/L=*line* Controls the line attaching the tag to the tagged point.

- line*=0: No line from tag to attachment point.
- line*=1: Line connecting tag to attachment point.
- line*=2: Line with arrow pointing from tag to attachment point.
- line*=3: Line with arrow pointing from attachment point to tag.
- line*=4: Line with arrows at both ends.

Tag

/LS= <i>linespace</i>	Specifies a tweak to the normal line spacing where <i>linespace</i> 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[= <i>sameSize</i>]	/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= <i>name</i>	Specifies name of the tag to create or change.
/O= <i>rot</i>	Sets the text's rotation. <i>rot</i> 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. If the tag is attached to a trace (not an image or axis), any non-zero /AO value will overwrite this rotation value.
/P= <i>tipOffset</i>	Sets the offset from the tip of a tag's line or arrow to the point on the wave that it is tagging. <i>tipOffset</i> is a positive number from 0 to 200 in points. If <i>tipOffset</i> =0 (default), it automatically chooses an appropriate offset.
/Q[= <i>contourInstance</i>]	Associates a tag with a particular contour level trace in a graph recreation macro. Of interest mainly to hard-core programmers. When “= <i>contourInstance</i> ” is present, /Q associates the tag with the contour wave. Igor will feel free to change or delete the tag, as appropriate, when it recalculates the contour (because you changed the contour data or appearance, the graph size or the axis range). <i>contourInstance</i> is a contour instance name, such as zWave or zWave#1 if you have the same wave contoured twice in the graph. /Q by itself, with “= <i>contourInstance</i> ” not present, disassociates the tag from the contour wave. Igor will no longer modify or delete the tag (unless the contour level to which it is attached is deleted). If you manually tweak a contour label, using the Modify Annotation dialog, Igor uses this flag.
/R= <i>newName</i>	Renames the tag.
/S= <i>style</i>	Controls the tag frame style. <i>style</i> =0: Single frame. <i>style</i> =1: Double frame. <i>style</i> =2: Triple frame. <i>style</i> =3: Shadow frame.
/T= <i>tabSpec</i>	<i>tabSpec</i> 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}.
/TL= <i>extLineSpec</i>	Specifies extended tag line parameters similar to the SetDrawEnv arrow settings. <i>extLineSpec</i> = { <i>keyword</i> = <i>value</i> ,...} or zero to turn off all extended specifications.

Valid keyword-value pairs are:

<code>len=l</code>	Length of arrow head in points ($l=0$ for auto).
<code>fat=f</code>	Width to length ratio of arrow head (default is 0.5 same as $f=0$).
<code>style=s</code>	Sets barb side mode (see SetDrawEnv astyle for values).
<code>shar=s</code>	Sets sharpness between -1 and 1 (default is 0; blunt).
<code>frame=f</code>	Sets frame thickness in outline mode.
<code>lThick=l</code>	Sets line thickness in points (default is 0.5 for $l=0$).
<code>lineRGB=(r,g,b[,a])</code>	Sets color for lines. r , g , b , and a specify the color and optional opacity as RGBA Values . The default is opaque black.
<code>dash=d</code>	Specifies dash pattern number between 0 and 17 (see SetDashPattern for patterns).

`/W=winName`

Operates on 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.

`/V=vis`

Controls annotation visibility.

- `vis=0`: Invisible annotation; not selectable. The annotation is still listed in **AnnotationList**.
- `vis=1`: Visible annotation (default).

`/X=xOffset`

Distance from point to tag as percentage of graph width. For axis tags, the offsets are proportional to the size of the text used for the axis labels.

`/Y=yOffset`

Distance from point to tag as percentage of graph height. For axis tags, the offsets are proportional to the size of the text used for the axis labels.

`/Z=freeze`

Controls freezing of tag position.

- `freeze=1`: Freezes tag position (you can't move it with the mouse).
- `freeze=0`: Unfreezes it.

Details

If the `/C` flag is used, it must be the first flag in the command 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 and must be followed immediately by the `/N=name` flag with no further flags or parameters.

taggedObjectName and *xAttach* can be omitted only when changing an existing tag using `Tag/C/N=<tag name>`. *textStr* can be included only if *taggedObjectName* and *xAttach* are included. This syntax allows changes to the tag to be made through the *flags* parameters without needing to respecify the other parameters. Similarly, the tag's attachment point can be changed without needing to respecify the *textStr* parameter.

A tag 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.

"\r" inserts a carriage-return character which starts a new line of text in the annotation.

Some escape codes insert text based on the wave point or axis to which a tag is attached. See **Tag Escape Codes** on page III-55 and **Axis Label Escape Codes** on page III-57 for details.