

AxisValFromPixel

AxisValFromPixel(*graphNameStr*, *axNameStr*, *pixel*)

The AxisValFromPixel function returns an axis value corresponding to the local graph pixel coordinate in the graph window or subwindow.

Parameters

graphNameStr can be "" to refer to the top graph window.

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

If the specified axis is not found and if the name is "left" or "bottom" then the first vertical or horizontal axis will be used. Sources for *pixel* value may be the GetWindow operation or a user window hook with the mousemoved and mousedown event messages (see the **SetWindow** operation).

If *graphNameStr* references a subwindow, *pixel* is relative to top left corner of base window, not the subwindow.

Axis ranges and other graph properties are computed when the graph is redrawn. Since automatic screen updates are suppressed while a user-defined function is running, if the graph was recently created or modified, you must call DoUpdate to redraw the graph so you get accurate axis information.

See Also

The **PixelFromAxisVal** and **TraceFromPixel** functions; the **GetWindow** and **SetWindow** operations.

BackgroundInfo

BackgroundInfo

The BackgroundInfo operation returns information about the current unnamed background task.

BackgroundInfo works only with the unnamed background task. New code should use named background tasks instead. See **Background Tasks** on page IV-319 for details.

Details

Information is returned via the following variables:

<i>V_flag</i>	0: No background task is defined. 1: Background task is defined, but not running (is idle). 2: Background task is defined and is running.
<i>V_period</i>	DeltaTicks value set by CtrlBackground. This is how often the background task runs.
<i>V_nextRun</i>	Ticks value when the task will run again. 0 if the task is not scheduled to run again.
<i>S_value</i>	Text of the numeric expression that the background task executes, as set by SetBackground.

See Also

The **SetBackground**, **CtrlBackground**, **CtrlNamedBackground**, **KillBackground**, and **SetProcessSleep** operations, and the **ticks** function. See **Background Tasks** on page IV-319 for usage details.

Base64Decode

Base64Decode(*inputStr*)

The Base64Decode function returns a decoded copy of the Base64-encoded string *inputStr*. The contents of *inputStr* are not checked for validity. Any invalid characters in *inputStr* are skipped, and decoding continues with subsequent characters.

The algorithm used to encode Base64-encoded data is defined in RFC 4648 (<http://www.ietf.org/rfc/rfc4648.txt>).

For an explanation of Base64 encoding, see <https://en.wikipedia.org/wiki/Base64>.

The Base64Decode function was added in Igor Pro 8.00.