If setting the scaling of any dimension (*x, y, z,* or *t*), *num1* is the starting index value — the scaled index for the first point in the dimension. The meaning of *num2* changes depending on the /I and /P flags. If you use /P, then *num2* is the delta value — the difference in the scaled index from one point to the next. If you use /I, *num2* is the “ending value” — the index value for the last element in the dimension. If you use neither flag, *num2* is the “right value” — the index value that the element *after the last element in the dimension* would have.

These three methods are just three different ways to specify the two scaling values, the starting value and the delta value, that are stored for each dimension of each wave.

If setting the data full scale (*d*), then *num1* is the nominal minimum and *num2* is the nominal maximum data value for the waves. The data full scale values are not used. They serve only to document the minimum and maximum values the waves are expected to attain. No flags are used when setting the data full scale.

The *unitsStr* parameter is a string that identifies the natural units for the x, y, z, t, or data values of the named waves. Igor will use this to automatically label graph axes. This string must be one to 49 bytes such as “m” for meters, “g” for grams or “s” for seconds. If the waves have no natural units you can pass `""` for this parameter.

Setting *unitsStr* to "dat" (case-sensitive) tells Igor that the wave is a date/time wave containing data in Igor date/time format (seconds since midnight on January 1, 1904). Date/time waves must be double-precision.

Flags

At most one flag is allowed, and then only if dimension scaling (not data full scale) is being set:

/I	Inclusive scaling. *num2* is the ending index — the index value for the very last element in the dimension.
/P	Per-point scaling. *num2* is the delta index value — the difference in scaled index value from one element to the next.

Details

SetScale will not allow the delta scaling value to be zero. If you execute a SetScale command with a delta value of zero, it will set the delta value to 1.0.

If you do not use the /P flag, SetScale converts *num1* and *num2* into a starting index value and a delta index value. If you call SetScale on a dimension with fewer than two elements, it does this conversion as if the dimension had two elements.

Prior to Igor Pro 3.0, Igor supported only 1D waves. “SetScale x” was used to set the scaling for the rows dimensions and “SetScale y” was used to set the data full scale. With the addition of multidimensional waves, “SetScale y” is now used to set the scaling of the columns dimension and “SetScale d” is used to set the data full scale. For backward compatibility, “SetScale y” on a 1D wave sets the data full scale.

When setting the dimension scaling of a numeric wave, you can omit the *unitsStr* parameter. Igor will set the wave’s scaling but not change its units. However, when setting the dimension scaling of a text wave, you must supply a *unitsStr* parameter (use `""` if the wave has no units). If you don’t, Igor will think that the text wave is the start of a string expression and will attempt to treat it as the *unitsStr*.

See Also

See Also

CopyScales, **DimDelta**, **DimOffset**, **DimSize**, **WaveUnits**

For an explanation of waves and dimension scaling, see **Changing Dimension and Data Scaling** on page II-68.

For further discussion of how Igor represents dates, see **Date/Time Waves** on page II-85.

SetVariable

SetVariable [/Z] ***ctrlName*** **[*keyword* = *value* [, *keyword* = *value* …]]**

The SetVariable operation creates or modifies a SetVariable control in the target window.

A SetVariable control sets the value of a global numeric or string variable or a point in a wave when you type or click in the control. A SetVariable can also hold its own value without the need for a global or wave.

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

Parameters

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

The following keyword=value parameters are supported:

activate

Activates the control and selects the text that sets the value. Use **ControlUpdate** to deactivate the control and deselect the text.

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 **Button** and **DefaultGUIControls** for more appearance details.

bodyWidth=*width*

Specifies an explicit size for the body (nontitle) portion of a SetVariable control. By default (bodyWidth=0), the body portion is the amount left over from the specified control width after providing space for the current text of the title portion. If the font, font size or text of the title changes, then the body portion may grow or shrink. If you supply a bodyWidth>0, then the body is fixed at the size you specify regardless of the body text. This makes it easier to keep a set of controls right aligned when experiments are transferred between Macintosh and Windows, or when the default font is changed.

disable=*d*

Sets user editability of the control.

d=0: Normal.
d=1: Hide.
d=2: No user input.

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

Sets the initial color of the title. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**. The default is opaque black.

To further change the color of the title text, use escape sequences as described for title=*titleStr*.

focusRing=*fr*

Enables or disables the drawing of a rectangle indicating keyboard focus:

fr=0: Focus rectangle will not be drawn.
fr=1: Focus rectangle will be drawn (default).

On Macintosh, regardless of this setting, the focus ring appears if you have enabled full keyboard access via the Shortcuts tab of the Keyboard system preferences.

font="*fontName*"

Sets the font used to display the value of the variable, e.g., `font="Helvetica"`.

format=*formatStr*

Sets the numeric format of the displayed value, e.g., format="%g". Not used with string variables. Never use leading text or the "%W" formats, because Igor reads the value back without interpreting the units. For a description of *formatStr*, see the **printf** operation.

frame=*f*

Sets the frame for the value readout.

f=0:	Value unframed.
f=1:	Value framed (default).

fsize=*s*

Sets the size of the type used to display the variable's value.

fstyle=*fs*

fs 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.

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

Specifies the background fill color for labels. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**. The default is 0, which uses the window's background color.

limits={*low*,*high*,*inc*}

Sets the limits of the allowable values (*low* and *high*) for the variable. *inc* sets the amount by which the variable is incremented if you click the control's up/down arrows. This applies to numeric variables, not to string variables. If *inc* is zero then the up/down arrows will not be drawn.

live=*l*

Determines when the readout is updated.

l=0:	Update only after variable changes (default).
l=1:	Update as variable changes.

noedit=*val*

noedit=1 prevents the user from clicking (or tabbing into) a SetVariable control to directly edit its value. This is useful when you want to make a string read-only or when you want to restrict a numeric setting to those available only via the control's up or down arrow buttons.

noedit=0 reactivates user editing.

noedit=2 is deprecated as of Igor Pro 6.34 but still supported. It allows the use of formatting escape codes described under **Annotation Escape Codes**. Use styledText=1, instead.

noproc

No procedure is to execute when the control's value is changed.

pos={*leftOrRight*,*top*}

Sets the position in **Control Panel Units** of the top/left corner of the control if its alignment mode is 0 or the top/right corner of the control if its alignment mode is 1. See the align keyword above for details.

pos+={*dx*,*dy*}

Offsets the position of the control in **Control Panel Units**.

proc=*procName*

Sets the procedure to execute when the control's value is changed.

rename=*newName*

Gives control a new name.

styledText=val

styledText=1 allows the use of formatting escape codes described under **Annotation Escape Codes** on page III-53. This works for string SetVariable controls only, not for numeric controls.

For example:

```
SetVariable sv0 value=_STR:"\\JC\\K(65535,0,0)Centered Red 
Text"
```

styledText=0 treats escape codes as plain text.

The styledText keyword was added in Igor Pro 6.34. For compatibility with earlier versions of Igor, the combination of noedit=1 and styledText=1 is recorded as noedit=2 in recreation macros.

size={*width*,*height*}
Sets width of control in **Control Panel Units**. *height* is ignored.

textAlign=*t*
Sets the alignment of the text displayed in the body of the control.

t=0:	Left (default)
t=1:	Center
t=2:	Right

The textAlign keyword was added in Igor Pro 9.00.

title=*titleStr*
Sets the title of the control to the specified string expression. The title is displayed to the left of the control. If *titleStr* is empty (`""`), the name of the controlled variable is displayed as the title. Use `title=" "` (put a space within the quotation marks) to create a "blank" title.

Using escape codes you can change the font, size, style, and color of the title. See **Annotation Escape Codes** on page III-53 or details.

userdata(*UDName*)=*UDStr*

Sets the unnamed user data to *UDStr*. Use the optional (*UDName*) to specify a named user data to create.

userdata(*UDName*)+=*UDStr*

Appends *UDStr* to the current unnamed user data. Use the optional (*UDName*) to append to the named *UDStr*.

value=*varOrWaveName*
Sets the numeric or string variable or wave element to be controlled.

If *varOrWaveName* references a wave, the point is specified using standard bracket notation with either a numeric point number or a row label, for example: `value=awave[4]` or `value=awave[%alabel]`.

You may also use a 2D, 3D, or 4D wave and specify a column, layer, and chunk index or dimension label in addition to the row index.

You can have the control store the value internally rather than in a global variable. In place of varName, use _STR:str or _NUM:num. For example:

```
NewPanel; SetVariable sv1,value=_NUM:123
```

valueColor=(*r*,*g*,*b*[,*a*])
Sets the color of the value text. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**. The default is opaque black.

valueBackColor=(*r*,*g*,*b*[, *a*])
Sets the background color under the value text. *r*, *g*, *b*, and *a* specify the color and optional opacity as **RGBA Values**.

valueBackColor=0
Sets the background color under the value text to the default color, the standard document background color used on the current operating system, which is usually white.

win=*winName*
Specifies which window or subwindow contains the named control. If not given, then the top-most graph or panel window or subwindow is assumed.

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