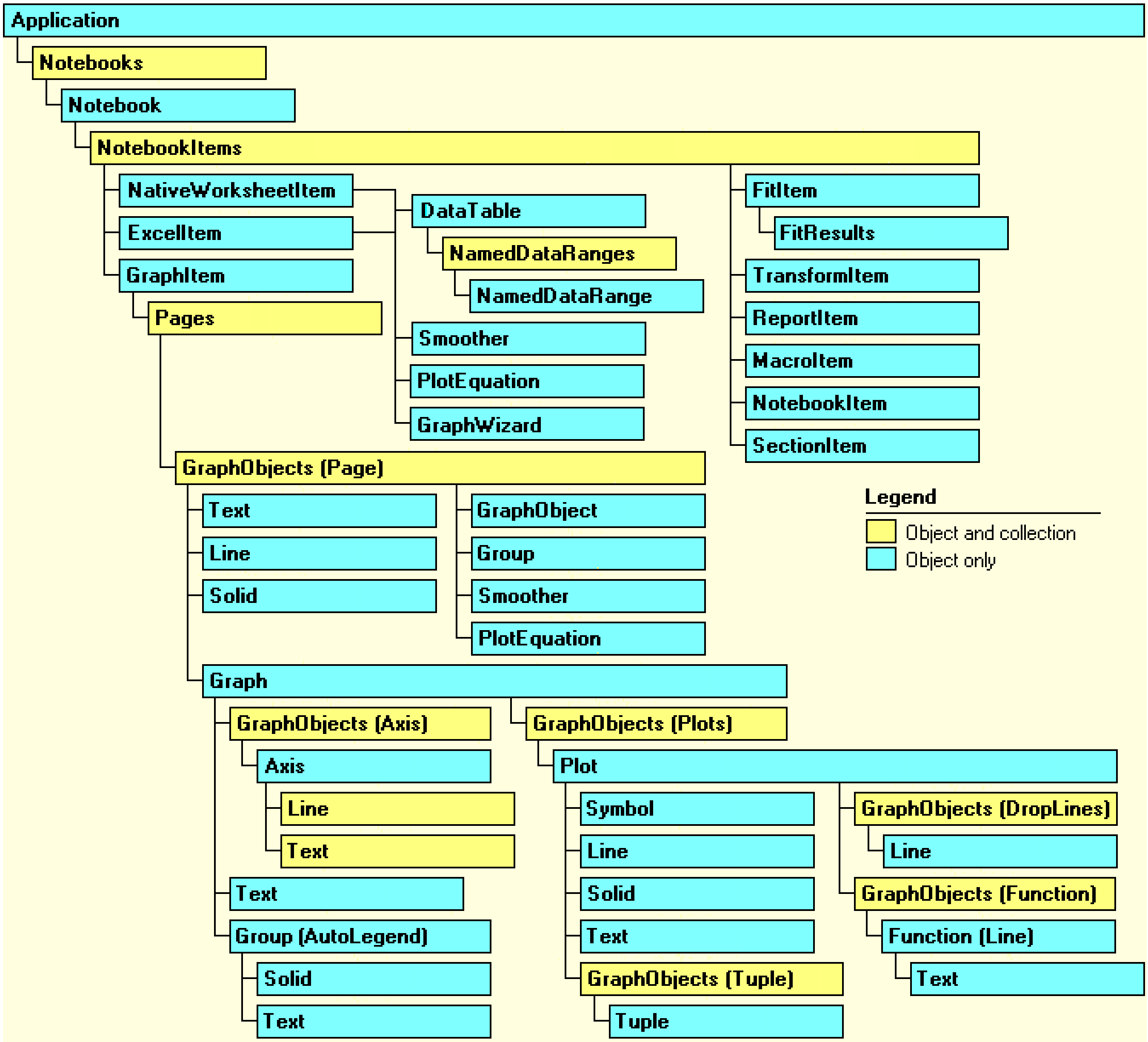
SigmaPlot Objects

About Objects



SigmaPlot Automation

SigmaPlot Objects

About Automation OLE Automation is a technology lets other applications, development tools, and macro languages use a program. SigmaPlot Automation allows you to integrate SigmaPlot with the applications you have developed. It also provides an effective tool to customize or automate frequent tasks you want to perform.

Automation uses objects to manipulate a program. Objects are the fundamental building block of macros; nearly all macro programs involve modifying objects. Every item in SigmaPlot—graphs, worksheets, axes, tick marks, reports, notebooks, etc.—can be represented by an object.

SigmaPlot uses a VBA®-like macro language to access automation internally. For more information on recording SigmaPlot macros, see Recording Macros

About Objects and Collections

Returning Objects About Properties About Methods

An *object* represents any type of identifiable item in SigmaPlot. Graphs, axes, notebooks, worksheets, and worksheet columns are all objects.

A *collection* is an object that contains several other objects, usually of the same type; for example, all the items in a notebook are contained in a single collection object. Collections can have methods and properties that affect the all objects in the collection.

Properties and methods are used to modify objects and collections of objects. To specify the properties and methods for an object that is part of a collection, you need to return that individual object from the collection first.

About Properties

About Objects and Collections About Methods

A property is a setting or other attribute of an object—think of a property as an "adjective." For example, properties of a graph include the size, location, type and style of plot, and the data that is plotted. To change the settings of an object, you change the properties settings. Properties are also used to access the objects that are below the current object in the hierarchy.

To change a property setting, type the object reference followed with a period, then type the property name, an equal sign (=), and the property value.

Example

```
Set Notebook.Title = "My Notebook"
```

Sets the name of the referenced SigmaPlot notebook to "My Notebook."

Note that some properties cannot be set, and only retrieved. The Help topic for each property indicates whether you can both set and retrieve that property (read-write), only retrieve the property (read-only), or only set the property (write-only).

You can get information about an object by returning the values of its properties.

Example

```
Set CurrentDoc = ActiveDocument.NotebookItems(3)
```

The fourth item in the current notebook (specified by ActiveDocument) is assigned to the variable CurrentDoc (item counts start with 0).

Properties List

About Methods

About Objects and Collections About Properties

Methods are an action that can be performed on or by an object—think of methods as "verbs." For example, the ExcelItem object has Copy and Clear methods. Methods can have parameters that specify the action ("adverbs").

Example

```
Notebooks(0).NotebookItems(2).Close(True)
```

This example closes the second item in the NotebookItems collection object while saving it first. Note that the NotebookItems collection is selected using the Notebooks object NotebookItems property.

Methods List

Returning Objects

Objects

In order to work with an object, you must be able to define the specific object by *returning* it. In general, most objects are returned using a property of the object above it in the object tree.

Returning Objects from Collections Other objects are returned by specifying a single object from a collection. Once you define the collection, you can return a specific object by using an index value (as you would with an array). You can use either the Item method shared by all collections, or use the index directly. The index can be the item name or a number. For example:

```
Set Worksheet = Notebooks("My Notebook").NotebookItems.Item(2)
```

The collection index value returns the notebook "My Notebook" from the Notebooks collection, then the Item property and index number returns the third item from the NotebookItems collection as the variable Worksheet.

The Notebooks collection contains a list of all the open notebooks in SigmaPlot, and the NotebookItems collection contains all items in the specified notebook.

Defining Variables Objects can also be returned and used by defining the object to be a variable, generally using the Dim (dimension) statement. Although you can implicitly declare variables just by using the variable for the first time, you can avoid bugs caused by typos using Option Explicit. For example, the script:

```
Option Explicit

Sub Main
Dim ItemCount
Dim SPWorksheets$()
ItemCount = ActiveDocument.NotebookItems.Count
ReDim SPWorksheets$(ItemCount)
Dim SPItems
Set SPItems = ActiveDocument.NotebookItems
Dim Index
Index = 0
Dim Item
For Each Item In SPItems
If SPItems(Index).ItemType = 1 Then
SPWorksheets$(Index) = SPItems(Index).Name
End If
Index = Index + 1
Next Item

Begin Dialog UserDialog 320,119,"Worksheets in Active Notebook" ' %GRID:10,7,1,1

OKButton 210,14,90,21
ListBox 20,14,170,91,SPWorksheets(),.ListBox1
End Dialog
Dim dlg As UserDialog
Dialog dlg
End Sub
```

Uses the Dim (Dimension) statement to define several variables, and uses the Set instruction to define a declared variable as an object.

Getting Help on Objects, Methods, and Properties

About Objects and Collections About Properties About Methods

Help Use Help to view the properties and methods for any object. Each object topic in Help includes Properties and Methods buttons that displays lists of the object's properties and methods. Press F1 in the Macro Window or Object Browser to jump to the appropriate Help topic.

Object Tree Displays SigmaPlot objects arranged in a tree format. Click an object to display the corresponding Help topic.

Object Browser The Object Browser in the Macro Window displays the members (properties and methods) of the SigmaPlot objects.

Searching Automation Help

SigmaPlot Automation Help offers three tools to assist in finding desired information. Each tool corresponds to a tab of the Help Topics dialog box.

- **Contents**. An outline of Automation Help, with topics grouped into meaningful categories.
- **Index**. An alphabetical list of Automation Help terms.
- **Find**. A full-text search through the Automation Help topics. This is particularly useful for finding constants recorded by the Macro Recorder.

The index and full-text search relate only to SigmaPlot Automation and Basic topics.

SigmaPlot Properties

About Properties

For Fit Item or FitResult Properties, see FitItem and FitResults Properties and Methods

DropLines

Expanded

Fill

FullName

Functions

Gallery

Graphs

GraphPages

Height

InsertionMode

Interactive

IsCurrentBrowserEntry

IsCurrentItem

IsOpen

ItemType

Keywords

Left

Line

LineAttributes

LowerPickIndex

Name

NameObject

NameOfRange

NamedRanges

NotebookItems

Notebooks

NumberFormat

ObjectType

OwnerGraphObject

Parent

Path

Plots

Saved

SelectedText

SelectionExtent

ShowStatsWorksheet

StatsWorksheetDataTable

StatusBar

Subject

SuspendIdle

Symbols

Template

Text

TickLabelAttributes

Title

Top

UpperPickIndex

Visible

Width

ActiveDocument Property

Objects

Read-Only
Value: Object

Syntax: ActiveDocument

Returns the active notebook (the notebook window in focus) as an object. If there are no notebooks open or if there is no document with the specified index, an error occurs and the value is returned as NULL.

To make a specific notebook the active document, use the Activate method.

Examples

```
ActiveDocument.Author = "John Doe"
ActiveDocument.Title = "My Notebook"
ActiveDocument.Comments = "For My Eyes Only"
```

Sets the Author, Title, and Descriptions fields of the summary information for the notebook item.

```
MsgBox ActiveDocument.FullName
```

Returns and displays the file name and path for the current notebook.

AddOnLocation Property

Objects

Read Only
Value: String

Syntax: *Application object*.AddOnLocation(a*ddon name* *variant*, *version* *variant*)

Returns the location of a SigmaPlot add-on or module from the Windows registry.

Example

```
Dim EKPath$
EKPath = AddOnLocation("Enzyme Kinetics")

MsgBox EKPath
```

Displays the path for the SigmaPlot Enzyme Kinetics Module.

Application Property

Objects

Read Only
Value: Object

Syntax: *object*.Application

Used without an object qualifier, this property returns an Application object that represents the SigmaPlot application. Used with an object qualifier, this property returns an Application object that represents the creator of the specified object (you can use this property with an Automation object to return that object's application).

Note: Use the CreateObject and GetObject functions to gain access to an Automation object.

Examples

```
MsgBox ActiveDocument.Application.FullName
```

Returns the name of the SigmaPlot executable file.

```
Set ActiveDocument.Application.DefaultPath = "c:\My Documents"
MsgBox ActiveDocument.Application.DefaultPath
```

Sets the default open and save path for the application to C:\My Documents.

Author Property

Objects

Read/Write
Value: String

Syntax: *Notebook/NotebookItems object*.Author

A standard property of notebook files and all NotebookItems objects. Returns or sets the Author field in the Summary Information for all notebook items, or the Author field under the Summary tab of the Windows 95/98 file Properties dialog box.

Examples

```
ActiveDocument.Author = "John Doe"
```

Changes the author of the current notebook to "John Doe."

```
MsgBox Notebooks(2).NotebookItems(3).Author
```

Returns and displays the author for the fourth item in the third open notebook.

AutoLegend Property

Objects

Read Only
Value: Object

Syntax: *Graph object*.AutoLegend

Returns the AutoLegend Group object for the specified Graph object. AutoLegends have all standard group properties. The first ChildObject of a legend is always a solid; the successive objects are text objects with legend symbols.

Examples

```
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).AutoLegend.ChildObjects(0).Color(RGB_YELLOW)
```

Changes the legend background color to yellow.

```
Dim SPLegend, Index
Set SPLegend = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).AutoLegend

Index = 0
Do While Index <= SPLegend.ChildObjects.Count – 2
SPLegend.ChildObjects(Index + 1).Name = "Curve " + CStr(Index + 1)
Index = Index + 1
Loop
```

Changes the names of all the legend labels to Curve *n*.

Axes Property

Objects

Read Only
Value: Object

Syntax: *Graph object*.Axes

The Axes property is used to return the collection of Axis objects for the specified graph object. Individual axis objects have a number of line and text objects that are returned with Axis object properties.

Examples

```
Dim SPGraph As Object
Set SPGraph = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0)
MsgBox "Graph " + SPGraph.Name + ": " + SPGraph.Axes.Count + " Axes",,"Number of Axes"
```

Displays the number of axes for the first graph on the current page.

```
Dim SPXAxis, Min, Max
Set SPXAxis = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(0)

SPXAxis.SetAttribute(SAA_OPTIONS,SAA_FLAG_AUTORANGE Or FLAG_CLEAR_BIT)

Begin Dialog UserDialog 340,98,"X Axis Range" ' %GRID:10,7,1,1
OKButton 240,7,90,21
Text 20,14,90,14,"Minimum",.Text1
TextBox 120,11,90,21,.Minimum
Text 20,42,90,14,"Maximum",.Text2
TextBox 120,39,90,21,.Maximum
CancelButton 240,35,90,21
End Dialog
Dim dlg As UserDialog
If Dialog(dlg) = 0 Then 'Handles Cancel button
GoTo Finish
End If
Min = dlg.Minimum
Max = dlg.Maximum
SPXAxis.SetAttribute(SAA_FROMVAL,Min)
SPXAxis.SetAttribute(SAA_TOVAL,Max)
Finish:
```

Provides a dialog interface for setting the X axis range for the first graph on the current page.

AxisTitles Property

Objects

Read Only
Value: Object

Syntax: *Axis object*.AxisTitles

The AxisTitle property is used to return the collection of axis title Text objects for the specified Axis . Use the following index values to return the different titles. Note the specific title returned depends on the current axis dimension/direction selected.

0 Bottom/Left axis title
1. Right/Top axis title
2. Sub axis title (not currently shown)

3. Sub axis title (not currently shown)

Examples

```
Dim SPAxes As Object
Set SPAxes = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes

SPAxes(0).AxisTitles(0).Name = "Bottom X Axis Title"
SPAxes(1).AxisTitles(0).Name = "Left Y Axis Title"
```

Renames the bottom X and left Y axis titles of the first graph on the current page.

```
Dim SPYAxis As Object
Set SPYAxis = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(1)

SPYAxis.AxisTitles(0).SetAttribute(STA_ORIENTATION,0)
```

Sets the orientation of the left Y axis title to 0°.

Cell Property

Objects

Read/Write
Value: Variant

Syntax: *DataTable object*.Cell (*Column* As Long, *Row* As Long)

Returns or sets the value of a cell with the specified column and row coordinates for the current DataTable object.

Examples

```
MsgBox ActiveDocument.NotebookItems("Data 1").DataTable.Cell(0,0)
```

Returns the contents of the cell in column 1, row 1 of the data table for the "Data 1" worksheet of the current notebook.

```
Dim Counter
Counter = 0
Dim NumberOfCells
Dim Cells
Do Until Counter >=100
ActiveDocument.NotebookItems(2).DataTable.Cell(0,Counter) = Counter+1

Counter = Counter + 1
Loop
```

Sets the value of cells 1 through 100 in column one to increment from 1 to 100.

Note: The Cell property is not a fast data placing operation; the PutData method is a much faster operation and should be used to place large arrays of data.

ChildObjects Property

Objects

Read Only
Value: Object (Collection)

Syntax: *Page object*.ChildObjects

Used by all page objects that contain different sub-objects to return the collection of those objects. The objects returned by the ChildObjects property depend on the object type:

Object	**ChildObjects Returns:**
Page	Page GraphObjects
Graph	Plots
Plot	Tuples
Tuples	Tuple
Group (including Autolegends	all group objects

Examples

```
MsgBox ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Count,,"Number of Objects"
```

Displays the number of objects on the current page.

```
Dim SPTuples As Object
Dim TupleCol
Set SPTuples = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0).ChildObjects

MsgBox "Column Plotted: " + _
CStr(SPTuples(0).GetAttribute(SNA_DATACOL,TupleCol)+1),,"Tuple 1"
```

Displays the column plotted by the first tuple in the first graph of the current page.

Color Property

Objects

Read/Write
Value: Long

Syntax: *Page object/childobject*.Color

Gets or sets the color for all drawn page objects. Use the different color constants for the standard VGA color set:

RGB_BLACK	0	&H00000000
RGB_BLUE	16711680	&H00FF0000
RGB_CYAN	16776960	&H00FFFF00
RGB_DKBLUE	8388608	&H00800000
RGB_DKCYAN	8421376	&H00808000
RGB_DKGRAY	8421504	&H00808080
RGB_DKGREEN	32768	&H00008000
RGB_DKPINK	8388736	&H00800080
RGB_DKRED	128	&H00000080
RGB_DKYELLOW	32896	&H00008080
RGB_GRAY	12632256	&H00C0C0C0
RGB_GREEN	65280	&H0000FF00
RGB_PINK	16713995	&H00FF00FF
RGB_RED	255	&H000000FF
RGB_WHITE	16777215	&H00FFFFFF
RGB_YELLOW	65525	&H0000FFFF

Examples

```
ActiveDocument.CurrentPageItem.GraphPages(0).Color = RGB_DKBLUE
```

Sets the current page color to dark blue.

```
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0).Fill.Color = RGB_DKRED
```

Changes the fill color of the solid object for the plot of the first graph to dark red.

ColumnBorderThickness Method

Objects

Type: Property Get
Result: Long

Syntax: *NativeWorksheetItem*.BorderWidth(*column* long)

Returns the border thickness for the specified worksheet column.

Example

```
Dim BorderWidth
BorderWidth = ActiveDocument.CurrentDataItem.ColumnBorderThickness(1)

MsgBox BorderWidth
```

Comments Property

Objects

Read/Write
Value: String

Syntax: *Notebook/NotebookItems object*.Comments

A standard property of notebook files and all NotebookItems objects. Returns or sets the Description field in the Summary Information for all notebook items, or the Comments section under the Summary tab of the Windows 95/98 file Properties dialog box for notebook files.

Examples

```
ActiveDocument.Comments = " Research data for Project X"
```

Changes the comments of the current notebook.

```
MsgBox Notebooks(1).NotebookItems(0).Comments
```

Returns and displays the comments for the notebook item in the second open notebook.

Count Property

Objects

Read Only
Value: Long

Syntax: *collection*.Count

A property available to all collection objects that returns the number of objects within that collection.

Examples

```
MsgBox Notebooks.Count
```

Displays the number of open notebook files

```
Dim SPItems$()
ReDim SPItems$(ActiveDocument.NotebookItems.Count)
```

Creates an SPItems array variable that is the size of the number of items in the current notebook.

CurrentBrowserItem Property

Objects

Read Only
Value: Object

Syntax: *Notebook object*.CurrentBrowserItem

Returns an object expression representing the currently selected object in the browser view.

Example

```
Dim msgtext, savestatus
If ActiveDocument.CurrentBrowserItem.Saved=True Then
savestatus="No need to save this item."
Else
savestatus="Changes have been made since last save."
End If
msgtext="Current Item: " + ActiveDocument.CurrentBrowserItem.Name + vbCr + _

savestatus
MsgBox(msgtext,0+64,"Status")
```

Lists the currently selected notebook item and whether the item should be saved or not.

CurrentDataItem Property

Objects

Read Only
Value: Object

Syntax: Notebook object

.CurrentDataItemThe CurrentDataItem property returns the worksheet window in focus as an object. You must still use the ActiveDocument property to specify the currently active notebook.

Note that if a worksheet is not in focus an error is returned.

Examples

```
ActiveDocument.CurrentDataItem.Interpolate3DMesh(1,2,3)
```

Creates interpolated mesh data for columns 1, 2 and 3 and places them in the first empty column.

```
Dim CurrentWorksheet As Object
Set CurrentWorksheet = ActiveDocument.CurrentDataItem
Dim Column As Long, Row As Long
Column = 0
Row = 0
CurrentWorksheet.DataTable.GetMaxUsedSize(Column,Row)
```

```
MsgBox "Column " + CStr(Column) + " to row " + CStr(Row),,CurrentWorksheet.Name + " Range"
```

Displays the current worksheet name and data table range.

CurrentDateString Property

Objects

Read Only
Value: String

Syntax: *Application object***.**CurrentDateString(*DatePicture*)

Returns formatted text representing the current date. "*DatePicture*" is a format string containing the following codes.

Picture	Meaning
d	Day of month as digits with no leading zero for single-digit days.
dd	Day of month as digits with leading zero for single-digit days.
ddd	Day of week as a three-letter abbreviation.
dddd	Day of week as its full name.
M	Month as digits with no leading zero for single-digit months.
MM	Month as digits with leading zero for single-digit months.
MMM	Month as a three-letter abbreviation.
MMMM	Month as its full name
y	Year as last two digits, but with no leading zero for years less than 10.
yy	Year as last two digits, but with leading zero for years less than 10.
yyyy	Year represented by full four digits.
gg	Period/era string.

Use the format codes to construct a format picture string. If you use spaces to separate the elements in the format string, these spaces will appear in the same location in the output string. The letters must be in uppercase or lowercase as shown (for example, "dd", not "DD"). Characters in the format string that are enclosed in single quotation marks will appear in the same location and unchanged in the output string.

For example, to get the date string

"Wed, Aug 31 94"

use the following picture string:

"ddd',' MMM dd yy"

If no picture string is supplied, the user’s current regional settings are used.

Example

```
MsgBox(Application.CurrentDateString("MMMM d, yyyy"),0+64,"Today's Date")
```

Displays the date.

CurrentItem Property

Objects

Read Only
Value: Object

Syntax: *Notebook object*.CurrentItem

This property returns whatever notebook item currently has focus as an object. You must still use the ActiveDocument property to specify the currently active notebook.

Examples

```
Dim CurrentItem As Object, ItemName As String
Set CurrentItem = ActiveDocument.CurrentItem
ItemName = InputBox$("Rename Current Item","Notebook Item Name",CurrentItem.Name)

If CurrentItem.Name = ItemName Then
GoTo Finish
Else
CurrentItem.Name = ItemName
End If
Finish:
```

Opens an input box that allows you to rename the current notebook item. The following code displays the item type for the current notebook item:

```
Dim CurrentItem As Object, TypeOfItem$, ItemCode As Integer

ItemCode = ActiveDocument.CurrentItem.ItemType
Select Case ItemCode
Case 1
TypeOfItem = "SigmaPlot Worksheet"
Case 2
TypeOfItem = "Graph Page"
Case 3
TypeOfItem = "Section"
Case 4
TypeOfItem = "SigmaStat Report"
Case 5
TypeOfItem = "SigmaPlot Report"
Case 6
TypeOfItem = "Equation"
Case 7
TypeOfItem = "Notebook"
Case 8
TypeOfItem = "Excel Worksheet"
Case 9
TypeOfItem = "Transform"
Case 10
TypeOfItem = "Macro"
End Select
MsgBox "Current Item is a " + TypeOfItem,,"Current Item"
```

CurrentPageItem Property

Objects

Read Only
Value: Object

Syntax: *Notebook object*.CurrentPageItem

Returns the current graph page window as a GraphItem object. You must still use the ActiveDocument property to specify the currently active notebook.

If the current item in focus is not a page, an error is returned.

Examples

```
ActiveDocument.CurrentPageItem.ApplyPageTemplate("Scatter Plot")
```

Applies the page template "Scatter Plot" to the current page.

```
Dim CurrentPage
Set CurrentPage = ActiveDocument.CurrentPageItem
MsgBox "# items on page: " + CurrentPage.GraphPages(0).ChildObjects.Count,,"Page: " + CurrentPage.Name
```

Displays the number of objects found on the current page.

CurrentPageObject Property

Objects

Read Only
Value: Object

Syntax: *ObjectVar* = *GraphItem object*.CurrentPageObject(*ObjectType* variant)

Returns an object reference to the "current" graph object of type "*ObjectType*". Valid values for *ObjectType* include: GPT_PAGE, GPT_GRAPH, GPT_AXIS, GPT_PLOT, GPT_TUPLE, GPT_LINE and GPT_OBJECT. These objects are normally operated on by the SetCurrentObjectAttribute method.

Example

```
Dim xname,yname
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SLA_SELECTDIM, 1)

xname=ActiveDocument.CurrentPageItem.GraphPages(0).CurrentPageObject(GPT_AXIS).Name
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SLA_SELECTDIM, 2)

yname=ActiveDocument.CurrentPageItem.GraphPages(0).CurrentPageObject(GPT_AXIS).Name
Begin Dialog UserDialog 400,84,"Axis Titles" ' %GRID:10,7,1,1
Text 30,14,50,14,"X-Axis:",.Text1
Text 30,49,40,14,"Y-Axis:",.Text2
TextBox 90,14,170,21,.TextBox1
TextBox 90,49,170,21,.TextBox2
OKButton 310,14,70,21
End Dialog
Dim dlg As UserDialog
dlg.TextBox1=xname
dlg.TextBox2=yname
Dialog dlg
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SLA_SELECTDIM, 1)
ActiveDocument.CurrentPageItem.GraphPages(0).CurrentPageObject(GPT_AXIS).Name=dlg.TextBox1
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SLA_SELECTDIM, 2)

ActiveDocument.CurrentPageItem.GraphPages(0).CurrentPageObject(GPT_AXIS).Name=dlg.TextBox2
```

Allows the user to change both the x-axis and y-axis titles in a single dialog.

CurrentTimeString Property

Objects

Read Only
Value: String

Syntax: *Application object***.**CurrentTimeString(*TimePicture*)

Returns formatted text representing the current time. "*TimePicture*" is a format string containing the following codes.

Picture	Meaning
h	Hours with no leading zero for single-digit hours; 12-hour clock
hh	Hours with leading zero for single-digit hours; 12-hour clock
H	Hours with no leading zero for single-digit hours; 24-hour clock
HH	Hours with leading zero for single-digit hours; 24-hour clock
m	Minutes with no leading zero for single-digit minutes
mm	Minutes with leading zero for single-digit minutes
s	Seconds with no leading zero for single-digit seconds
ss	Seconds with leading zero for single-digit seconds
t	One character time marker string, such as A or P
tt	Multicharacter time marker string, such as AM or PM

Use the format codes to construct a format picture string. If you use spaces to separate the elements in the format string, these spaces will appear in the same location in the output string. The letters must be in uppercase or lowercase as shown (for example, "ss", not "SS"). Characters in the format string that are enclosed in single quotation marks will appear in the same location and unchanged in the output string.

For example, to get the time string

"11:29:40 PM"

use the following picture string:

"hh':'mm':'ss tt"

If no picture string is supplied, the user’s current regional settings are used.

Example

```
MsgBox(Application.CurrentTimeString("hh:mm"),0+64,"Current Time")
```

Displays the time.

Data Format Names

Simples (one curve) plots
XY Pair
Single X
Single Y

Multiple curve and column plots
XY Pairs
X Many Y
Y Many X
Many X
Many Y

Polar plots
ThetaR
XY Pairs
Theta Many R
R Many Theta
Many R
Many Theta

3D and contour
XYZ Triplet (not available for bar charts)
Many Z
XY Many Z

Ternary
Ternary Triplets
Ternary XY Pairs
Ternary YZ Pairs
Ternary XZ Pairs

Pie
Single Column

DataTable Property

Objects

Read Only
Value: Object

Syntax: *NativeWorksheetItem/ExcelItem/GraphItem object*.DataTable

Returns the DataTable object for the specified worksheet object.

Examples

```
Dim Data As Object
```

```
Set Data = ActiveDocument.NotebookItems("Data 1").DataTable
```

Declares and sets the Data variable to be the DataTable objects of the "Data 1" worksheet.

```
Dim X As Long
Dim Y As Long
ActiveDocument.NotebookItems(2).DataTable.GetMaxUsedSize(X,Y)

MsgBox CStr(X) + ", " + CStr(Y)
```

Displays the last column and row used in the current data table for the first worksheet.

DecimalSymbol Property

Objects

Read Only
Value: String

Syntax: DecimalSymbol

Returns the decimal symbol used in the Windows Regional Settings

Example

```
Dim DecimalChar$
DecimalChar = DecimalSymbol
MsgBox "Current Decimal Symbol: " + DecimalChar
```

Displays the current system decimal symbol.

DefaultPath Property

Objects

Read/Write
Value: String

Syntax: DefaultPath

Sets or returns the default path used by the Application object to save and retrieve files. Files are opened using the Notebooks collection Open method and saved using the Notebook object Save or SaveAs methods.

Examples

```
DefaultPath = "C:\My Documents"
```

Sets the path used to open and save notebook files to C:\My Documents

```
MsgBox DefaultPath
```

Displays the current default path.

DropLines Property

Objects

Read Only
Value: Object

Syntax: *Plot object*.DropLines

Returns the DropLines line collection for a Plot object. Line objects within the DropLines collection have standard line properties.

Use an index to return a specific set of drop lines from the DropLines colletion:

1. xy plane (SLA_FLAG_DROPZ , 3D graphs only)
2. Y axis/x direction or yz plane (SLA_FLAG_DROPX)
3. X axis/y direction or zx plane (SLA_FLAG_DROPY)

Some drop line properties are controlled from the Plot object; for example, use the SetAttribute (SLA_PLOTOPTIONS,SLA_FLAG_DROPX Or FLAG_SET_BIT) plot object method to turn on y axis drop lines. Other drop line properties are set using Line object attributes.

Examples

```
Dim SPPlot As Object, SPDropLines As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0)

Set SPDropLines = SPPlot.DropLines
SPPlot.SetAttribute(SLA_PLOTOPTIONS,SLA_FLAG_DROPZ Or FLAG_SET_BIT)
SPDropLines(1).Color = RGB_GRAY
```

Turns on the z-direction drop lines for a 3D graph and turns the drop line colors to gray.

```
Dim SPPlot As Object, SPDropLines As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0)

Set SPDropLines = SPPlot.DropLines
SPPlot.SetAttribute(SLA_PLOTOPTIONS,SLA_FLAG_DROPX Or FLAG_SET_BIT)
SPDropLines(3).SetAttribute(SEA_LINETYPE,SEA_LINE_DOTTED)
```

Turns on the drop lines to the Y axis and sets their line type to dotted.

Expanded Property

Objects

Read/Write
Value: Boolean

Syntax: *NotebookItem/SectionItem object*.Expanded

A property of notebook window notebooks and sections, which opens or closes the tree for that notebook section, or returns a true or false value for the current view.

Examples

```
MsgBox ActiveDocument.NotebookItems(1).Expanded
```

Displays the expanded status for the first section of the current notebook. Note that NotebookItems(1) always corresponds to the first section.

```
ActiveDocument.NotebookItems(0).Expanded = False
```

Closes the notebook tree for the current notebook.

Fill Property

Objects

Read Only
Value: Object

Syntax: *Plot object*.Fill

The Fill property is used to return the Solid object for the specified Plot object. Solid objects for plots include bars and boxes.

Examples

```
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0).Fill.Color = RGB_GRAY
```

Changes the fill color for the first plot to gray.

```
Dim SPPlot
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0)

SPPlot.Fill.SetAttribute(SDA_EDGECOLOR,RGB_RED)
```

Sets the border color of the solid object in the current plot to red.

FullName Property

Objects

Read Only
Value: String

Syntax: *Application/Notebook object*.FullName

Returns the filename and path for either the application or the current notebook object. If the notebook object has not yet been saved to a file, an empty string is returned.

Example

```
MsgBox ActiveDocument.FullName
```

Displays the path and filename used by the current notebook.

Functions Property

Objects

Read Only
Value: Object (Collection)

Syntax: *Plot object*.Functions

The Functions property is used to return the collection of Function objects for the specified Plot object. Plot functions include regression and confidence lines, and all reference (QC) lines. The individual function lines are specified using an index:

Index	Constant	Function
1.	SLA_FUNC_REGR	Regression Line
2.	SLA_FUNC_CONF1	Upper Confidence Intervals
3.	SLA_FUNC_CONF2	Lower Confidence Interval
4.	SLA_FUNC_PRED1	Upper Prediction Interval
5.	SLA_FUNC_PRED2	Lower Prediction Interval
6.	SLA_FUNC_QC1	1. st Reference Line (Upper Specification)
7	SLA_FUNC_QC2	2. nd Reference Line (Upper Control Line)
8	SLA_FUNC_QC3	3. rd Reference Line (Mean)
9	SLA_FUNC_QC4	4. th Reference Line (Lower Control Line)
10	SLA_FUNC_QC5	5. th Reference Line (Lower Specification)

Note that most regression and reference lines options are controlled with different plot and line attibutes. For example, to turn on a regression line, use SetAttribute (SLA_REGROPTIONS,SLA_REGR_FORPLOT Or FLAG_SET_BIT), and to turn on the third reference line, use SetAttribute(SLA_QCOPTIONS,SLA_QCOPTS_SHOWQC3 Or FLAG_SET_BIT)

Examples

```
Dim SPPlot As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0)
SPPlot.Functions(SLA_FUNC_REGR).SetAttribute(SEA_LINETYPE,SEA_LINE_DOTTED)

SPPlot.Functions(SLA_FUNC_REGR).Color = RGB_BLACK
```

Changes the line type to dotted and the color to black for the regression line on for the first plot of the current page.

```
Dim SPPlot As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0)

SPPlot.SetAttribute(SLA_QCOPTIONS,SLA_QCOPTS_SHOWQC3 Or FLAG_SET_BIT)

SPPlot.Functions(8).Color = RGB_RED
```

Turns on the Mean reference line for the first plot and sets the color to red.

Gallery Property

Objects

Read-Only
Value: Object

Syntax: Gallery

Returns the current Graph Style Gallery notebook as an object.

Example

```
Dim GalleryPages$()
Dim i As Integer
Dim Item As Object
i=0
For Each Item In Gallery.NotebookItems
If Item.ItemType = 2 Then
ReDim Preserve GalleryPages(i)
GalleryPages(i)=Item.Name
i=i+1
End If
Next Item
Begin Dialog UserDialog 480,203,"Gallery Pages" ' %GRID:10,7,1,1

OKButton 390,175,80,21
ListBox 10,28,460,140,GalleryPages(),.ListBox1
Text 10,7,460,14,Gallery.FullName,.Text1
End Dialog
Dim dlg As UserDialog
Dialog dlg
```

Displays the current gallery file and all styles available from the gallery.

GraphPages Property

Objects

Read Only
Value: Object

Syntax: *GraphItem object*.GraphPages

Returns the GraphPages collection of Page objects for a GraphItem object. However, since there is currently only one graph page for any given graph item, you can always use GraphPages(0). However, in order to access items within a GraphItem, you must always specify the GraphPage.

Example

```
Dim SPGraphPage As Object
Set SPGraphPage = ActiveDocument.CurrentPageItem.GraphPages(0)

MsgBox SPGraphPage.Graphs.Count,,"Number of Graphs"
```

Displays a count of the graphs in the default page.

Graphs Property

Objects

Read Only
Value: Object

Syntax: *Page object*.Graphs

Returns the collection of graphs for the specified Page object. Use the index to select a specific Graph object. Graphs are used to return the different graph items: Plots, Axes, the graph title, and the graph legend.

Examples

```
MsgBox ActiveDocument.CurrentPageItem.GraphPages(0).Graphs.Count,,"# Graph on Page"
```

Displays a count of the number of graphs on the current page.

```
Dim SPGraph As Object, GraphName$
Set SPGraph = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0)

GraphName = InputBox$("Rename Graph","Graph Title",SPGraph.Name)
If SPGraph.Name = GraphName Then
GoTo Finish
Else
SPGraph.Name = GraphName
End If
Finish:
```

Opens a dialog to rename the first graph of the current page.

Height Property

Objects

Read/Write
Value: Long

Syntax: *Notebook/NotebookItems document object*.height

Sets or returns the height of the application window or specified notebook document window in pixels, or the size of pages and page objects in 1000ths of an inch.

Examples

```
ActiveDocument.NotebookItems("Data 1").Height = 500
```

Sets the height of the "Data 1" notebook item window to 500.

```
Dim SPPage
Set SPPage = ActiveDocument.NotebookItems("Graph Page 1").GraphPages(0)

MsgBox ("Page Size is " + CStr(SPPage.Height/1000) + " in. x " + _
CStr(SPPage.Width/1000 + " in."),vbInformation,"Page Size")
```

Displays the height and width for "Graph Page 1."

Note: To set the sizes and position at once, use a With statement:

```
With Application

.Left = 0
.Top = 0
.Height = 600
.Width = 800
End With
```

InsertionMode Property

Objects

Read Only
Value: Boolean

Syntax: *NativeWorksheetItem*.InsertionMode

Sets or returns a Boolean indicating whether or not Insert mode is on. When Insert mode is on, a new cell entry shifts the entire column down by one cell. When Insert mode is off, a new cell entry overwrites the current cell contents.

Example

```
ActiveDocument.NotebookItems("Data 1").InsertionMode = True
```

Turns Insert mode on for the "Data 1" worksheet.

Interactive Property

Objects

Read/Write
Value: Boolean

Syntax: Interactive

Sets or returns a Boolean indicating whether or not the user is allowed to interact with the application. Exercise care when setting the Interactive property to False from within SigmaPlot; if the value is not True upon exit of the macro, you will lose access to the application.

Example

```
Dim SPApp As Object
Set SPApp = CreateObject("SigmaPlot.Application.1")

SPApp.Visible=True
SPApp.Interactive=False
```

Creates a SigmaPlot application object from VB or VBA, and makes SigmaPlot ignore all user actions within the application window. Note that by default, SigmaPlot is also hidden from view when automated from another application.

IsCurrentBrowserEntry Property

Objects

Read/Write
Value: Boolean

Syntax: Interactive

Returns whether or not the specified item is the currently selected item in the notebook tree. This is particularly useful when adding new objects to a notebook in a specific notebook location.

Example

```
ActiveDocument.NotebookItems("Native Worksheet").IsCurrentBrowserEntry = True

ActiveDocument.NotebookItems.Add(CT_GRAPHICPAGE)
ActiveDocument.NotebookItems("Excel Worksheet").IsCurrentBrowserEntry = True
ActiveDocument.NotebookItems.Add(CT_GRAPHICPAGE)
```

Adds two graph pages to the current notebook. The first graph page is added below the "Native Worksheet" item by making this worksheet the current item. The second graph page follows the "Excel worksheet item.

IsCurrentItem Property

Objects

Read/Write
Value: Boolean

Syntax: Interactive

Returns whether or not the specified item is the currently selected item. This property is particularly useful when used in conjunction with the CurrentItem property.

Example

```
Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim index2
index2=0
Dim DataList$(ActiveDocument.NotebookItems.Count)
Dim Item
For Each Item In ActiveDocument.NotebookItems
If ActiveDocument.NotebookItems(Index).IsOpen=True Then
If ActiveDocument.NotebookItems(Index).ItemType = 1 Or ActiveDocument.NotebookItems(Index).ItemType = 8 Then

DataList$(Index2) = ActiveDocument.NotebookItems(Index).Name
index2=index2+1
ActiveDocument.NotebookItems(Index).Open
End If
End If
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Open Worksheets in Active Notebook" ' %GRID:10,7,1,1
OKButton 210,14,90,21
ListBox 20,14,170,91,DataList(),.ListBox1
End Dialog
Dim dlg1 As UserDialog
Dialog dlg1

ActiveDocument.NotebookItems(DataList$(dlg1.ListBox1)).IsCurrentItem = True
Dim sourcecol As String
sourcecol=InputBox$("Which column do you want to copy?","Source Column","1")
Dim MaxColumn As Long
Dim MaxRows As Long
MaxColumn = 0
MaxRows = 0
ActiveDocument.NotebookItems(DataList$(dlg1.ListBox1)).DataTable.GetMaxUsedSize(MaxColumn,MaxRows)
Dim Column1() As Variant
Column1()=ActiveDocument.CurrentDataItem.DataTable.GetData(CLng(sourcecol)-1,0,CLng(sourcecol)-1,MaxRows-1)
Begin Dialog UserDialog 320,119,"Target Worksheets" ' %GRID:10,7,1,1
OKButton 210,14,90,21
ListBox 20,14,170,91,DataList(),.ListBox1
End Dialog
Dim dlg2 As UserDialog
Dialog dlg2

ActiveDocument.NotebookItems(DataList$(dlg2.ListBox1)).IsCurrentItem = True
ActiveDocument.CurrentItem.Open
sourcecol=InputBox$("In which column do you want to place the data?","Source Column","1")
ActiveDocument.CurrentDataItem.DataTable.PutData(Column1,CLng(sourcecol)-1,0)
```

Copies a specified column from a selected open worksheet and pastes the column into the specifed location in another open worksheet.

IsEmbeddedDoc Property

Objects

Read Only
Value: Boolean

Syntax: *Notebook object*.IsEmbeddedDoc

This property is used to determine if the specified notebook document is an OLE embedded document.

Example

```
MsgBox ActiveDocument.IsEmbeddedDoc
```

Displays whether or not active notebook is embedded in another document.

IsOpen Property

Objects

Read Only
Value: Boolean

Syntax: *NotebookItems object*.IsOpen

A property common to all NotebookItems objects. Returns a Boolean indicating whether or not the specified document or section is open. Open and close notebook items using the Open and Close methods.

Example

```
MsgBox ActiveDocument.NotebookItems(2).IsOpen
```

Displays whether or not the third item of the current notebook is open.

ItemType Property

Objects

Read Only
Value: Integer

Syntax: *NotebookItems object*.ItemType

A property common to all NotebookItems objects. Returns an integer denoting the item /object type.

1.	CT_WORKSHEET	NativeWorksheetItem
2.	CT_GRAPHICPAGE	GraphItem
3.	CT_FOLDER	SectionItem
4.	CT_STATTEST	ReportItem (SigmaStat)
5.	CT_REPORT	ReportItem (SigmaPlot)
6.	CT_FIT	FitItem
7.	CT_NOTEBOOK	NotebookItem
8.	CT_EXCELWORKSHEET	ExcelItem
9.	CT_TRANSFORM	TransformItem
1.		MacroItem

Example

The following macro lists all notebook items by number in a dialog, then returns the item type as a string mapped to the ItemType property code.

```
Dim Items$()
ReDim Items(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
While Index<= ActiveDocument.NotebookItems.Count
Items$(Index) = CStr(Index)
Index = Index + 1
Wend
Begin Dialog UserDialog 250,154,"Select the Item Number" ' %GRID:10,7,1,1
OKButton 150,14,90,21
ListBox 10,14,110,126,Items(),.ItemNumber
End Dialog
Dim dlg As UserDialog
Dialog dlg
Dim ItemTypeName$
Select Case ActiveDocument.NotebookItems(CLng(dlg.ItemNumber)).ItemType
```

```
Case 1
ItemTypeName$ = "SigmaPlot Worksheet"
Case 2
ItemTypeName$ = "Graphics Page"
Case 3
ItemTypeName$ = "Section"
Case 4, 5
ItemTypeName$ = "Report"
Case 6
ItemTypeName$ = "Equation"
Case 7
ItemTypeName$ = "Notebook"
Case 8
ItemTypeName$ = "Excel Worksheet"
Case 9
ItemTypeName$ = "Transform"
Case 10
ItemTypeName$ = "Macro"
Case Else
ItemTypeName$ = "No Item"
End Select

MsgBox "The item type is "+ItemTypeName$
```

Keywords Property

Objects

Read/Write
Value: String

Syntax: *Notebook/NotebookItems object*.Keywords

A standard property of notebook files and all NotebookItems objects. Sets the Keywords field under the Summary tab of the Windows 95/98 file Properties dialog box.

Note that the keywords for notebook items are not currently displayed or used. The default keywords used by SigmaPlot notebooks are "SigmaPlot" and "SigmaStat."

Examples

```
ActiveDocument.Keywords = "Project X"
```

Changes the keywords of the current notebook to "Project X."

```
MsgBox Notebooks(0).Keywords
```

Returns and displays the keywords used for the first open notebook.

Left Property

Objects

Read/Write
Value: Long

Syntax: *Notebook/NotebookItems document object*.Left

Sets or returns the left coordinate of the application window or specified notebook document window in pixels, or the size of pages and page objects in 1000ths of an inch.

Examples

```
ActiveDocument.NotebookItems("Data 1").Left = 0
```

Sets the left side of the "Data 1" notebook item window to 0.

```
MsgBox ActiveDocument.NotebookItems("Graph Page 1").Left
```

Returns the left coordinate of the "Graph Page 1" notebook item.

Note: To set the window size and position at once, use a With statement:

```
With Application

.Left = 0
.Top = 0
.Height = 600
.Width = 800
End With
```

Line Property

Objects

Read Only
Value: Object

Syntax: *Plot object*.Line

Returns the Line object for the specified Plot object. Lines are available in both line plots and line and scatter plots.

Example

```
Dim SPLine As Object
Set SPLine = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0).Line

SPLine.SetAttribute(SEA_THICKNESS,50)
SPLine.Color = RGB_DKRED
```

Changes the line color for the first plot to dark red and the line thickness to 0.05 inches.

LineAttributes Property

Objects

Read Only
Value: Object

Syntax: *Axis object*.LineAttributes

Returns the collection of axis Line objects for the specified Axis object. Use the collection index to return a specific line object:

Index	Line
1.	Axis Lines
2.	Major Ticks
3.	Minor Ticks
4.	Major Grid
5.	Minor Grid
6.	Axis Break

Note that many axis line attributes are set with the different Axis object attributes, using the Axis object SetAttribute method.

Example

```
Dim SPHoriz,SPVert
Set SPHoriz = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(0).LineAttributes(1)

Set SPVert = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(1).LineAttributes(1)
SPHoriz.Color(RGB_BLUE)
SPVert.Color(RGB_RED)
Set SPHoriz = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(0).LineAttributes(4)

Set SPVert = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(1).LineAttributes(4)
SPHoriz.SetAttribute(SEA_LINETYPE,6)
SPVert.SetAttribute(SEA_LINETYPE,6)
SPHoriz.Color(RGB_GRAY)
SPVert.Color(RGB_GRAY)
Dim i,breakstatus,brkparam(2)
For i=0 To 1
Set SPHoriz = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(i)
breakstatus=SPHoriz.GetAttribute(SAA_BREAKON,brkparam(i))
If breakstatus=1 Then
SPHoriz.LineAttributes(6).Color(RGB_BLACK)
SPHoriz.SetAttribute(SAA_BREAKTYPE,2)
SPHoriz.LineAttributes(6).SetAttribute(SEA_LINETYPE,6)
End If
Next i
```

Changes the horizontal axis lines to blue and the vertical axis lines to red. Gridlines for both axes are set to a gray, dotted style. In addition, if either axis contains a break, the break appears as two black, diagonal, dotted, parallel lines.

ListSeparator Property

Objects

Read Only
Value: String

Syntax: ListSeparator

Returns the list separator symbol from the Windows Regional Settings.

Example

```
Dim SeparatorChar$
SeparatorChar = ListSeparator
MsgBox "Current Separator Symbol: " + SeparatorChar
```

Displays the current list separator symbol used by the Windows Regional Settings.

LowerPickIndex Property

Objects

Read Only
Value: Long

Syntax: *GraphWizard object*.LowerPickIndex

Returns the lower range of the index(s) picked by the graph wizard. See also UpperPickIndex

The lower index is the first column picked to plot for the graph created by the finishing of the GraphWizard object. These values are not correctly initialized until the graph wizard has run to completion.

The upper and lower indexes correspond to the indexes data titles set by the SetTitles method.

See the GraphWizard object for examples of using the upper and lower index values.

Name Property

Objects

Read/Write
Value: String

Syntax: *Notebook/NotebookItems object*.Name

A standard property of almost all SigmaPlot objects. Returns or sets the Title name and field in the Summary Information for all notebook items, the filename for a notebook file, and the object name or title for page objects.

To set the title used for a notebook, use the Notebook object Title property, or set the name for NotebookItems(0).

Note: If you attempt to set the name of a document to the existing name, you will receive an error message and the macro will halt.

Examples

```
ActiveDocument.NotebookItems(0).Name = "Project X Notebook"
```

Changes the comments of the current notebook.

```
MsgBox ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Name,,"Graph Title"
```

Returns and displays the title/name for the first graph of the current page.

NamedRanges Property

Objects

Read Only
Value: Object (collection)

Syntax: *DataTable object*.NamedRanges

Returns the collection of NamedDataRanges from a DataTable object. Use the NamedDataRanges collection to return a specific NamedDataRange object.

Examples

```
Dim Data1Ranges
Set Data1Ranges = ActiveDocument.NotebookItems("Data 1").DataTable.NamedRanges
```

Declares and sets the variable Data1Range to be the collection of named data ranges in the Data 1 worksheet.

```
MsgBox Notebooks(0).NotebookItems("Data 1").DataTable.NamedRanges(0).NameOfRange
```

Displays the name of the first named range in the NamedDataRange collection.

NameObject Property

Objects

Read Only
Value: Object

Syntax: *Page child object*.NameObject

Returns the Text object that corresponds to the name of the specified object.

Example

```
Dim SPAxis
Set SPAxis = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(1)
Dim newtitle As String
newtitle = SPAxis.Name + " vs. "
Set SPAxis = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(0)
newtitle = newtitle + SPAxis.Name
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).NameObject.Name = newtitle
```

Retitles the plot using the current x and y axis labels.

NameOfRange Property

Objects

Read/Write
Value: String

Syntax: *NamedDatRange object*.NameOfRange

Sets or returns the name for a NamedDataRange object. Useful for returning lists of column and row titles, which are named ranges.

Example

The following example retrieves the NamedDataRanges collection from the Data 1 worksheet in the current notebook, then lists them by name.

```
Dim NamedRangeArray$()
Dim SPRanges
Set SPRanges = ActiveDocument.NotebookItems("Data 1").DataTable.NamedRanges

ReDim NamedRangeArray$(SPRanges.Count)
Dim Index
Index = 0
Dim Item
For Each Item In SPRanges
NamedRangeArray$(Index) = SPRanges(Index).NameOfRange
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Named Ranges in Data 1" ' %GRID:10,7,1,1
OKButton 210,14,90,21
ListBox 20,14,170,91,NamedRangeArray(),.ListBox1
End Dialog
Dim dlg As UserDialog
Dialog dlg
```

NotebookItems Property

Objects

Read Only
Value: Object (collection)

Syntax: *Notebook object*.NotebookItems

A Notebook object property that returns the collection of notebook items. Use the NotebookItems collection to access individual notebook items. Worksheets, pages, equations, reports, macros, and section and notebook folders are all notebook items and can be returned as objects.

Example

This example lists all the notebook items found in the current notebook by name.

```
Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim Item
For Each Item In ActiveDocument.NotebookItems
NotebookItems$(Index) = ActiveDocument.NotebookItems(Index).Name
Index = Index + 1
```

```
Next Item
Begin Dialog UserDialog 320,119,"Items in Active Notebook" ' %GRID:10,7,1,1

OKButton 210,14,90,21
ListBox 20,14,170,91,NotebookItems(),.ListBox1
End Dialog
Dim dlg As UserDialog
Dialog dlg
```

Notebooks Property

Objects

Read Only
Value: Object (collection)

Syntax: Notebooks

An Application object property that returns the Notebooks collection object. Use the Notebooks collection to return individual Notebook objects and create new notebooks.

Example

The following script retrieves all notebooks and displays them by title. Note that the Title property displays the NotebookItem name, whereas the Name property returns the filename, which is not created until the notebook is saved.

```
Dim NotebookList$()
ReDim NotebookList$(Notebooks.Count)
Dim Index
Index = 0
Dim Item
For Each Item In Notebooks
NotebookList$(Index) = Notebooks(Index).Title
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Open Notebook List" ' %GRID:10,7,1,1

OKButton 210,14,90,21
ListBox 20,14,170,91,NotebookList(),.ListBox1
End Dialog
Dim dlg As UserDialog
Dialog dlg
```

NumberFormat Property

Objects

Read/Write
Value: String

Syntax: *NativeWorksheetItem object*.NumberFormat

Sets or returns the format used by the currently selected cells in the DataTable of the NativeWorksheetItem or ExcelItem object. If there is no selection, the format for the entire worksheet is assumed. If there are mixed formats, a NULL value is returned.

Both Number and Date and Time formats are set or returned using the standard number and date and time format designations.

Examples

```
MsgBox ActiveDocument.NotebookItems("Data 1").NumberFormat
```

Returns the format used by the currently selected cells in the worksheet "Data 1."

```
Notebook(0).NotebookItems(2).NumberFormat = "0.000[E+00]"
Notebook(0).NotebookItems(2).NumberFormat = "MMMM d, yyyy"
```

Sets the number format for the selected worksheet to three decimal places, and the date format to a long date (e.g. January 1, 1999).

ObjectType Property

Objects

Read Only
Value: Long

Syntax: *Page object/child object*.ObjectType

Returns the type value for the specified object. The values returned and corresponding object types are:

Value	Constant	Object
1.	GPT_PAGE	Page
2.	GPT_GRAPH	Graph
3.	GPT_PLOT	Plot
4.	GPT_AXIS	Axis
5.	GPT_TEXT	Text
6.	GPT_LINE	Line
7.	GPT_SYMBOL	Symbol
8.	GPT_SOLID	Solid
9.	GPT_TUPLE	Tuple
1.	GPT_FUNCTION	Function
1.	GPT_EXTERNAL	GraphObject
1.	GPT_BAG	Group
14	GPT_DATATABLE	DataTable

OwnerGraphObject Property

Objects

Read Only
Value: Object

Syntax: *Page child object*.OwnerGraphObject

Returns the object that the current object is contained within. This applies to the different graph page object hierarchies, where the Parent property is not supported.

Example

```
MsgBox ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0).OwnerGraphObject.Name
```

Returns the name of the first graph on the current page.

Parent Property

Objects

Read Only
Value: Object

Syntax: *object*.Parent

Returns the object or collection immediately "above" the current object. For graph page items, use the OwnerGraphObject property instead.

Example

```
Dim SPItem
Set SPItems = ActiveDocument.NotebookItems

MsgBox SPItem.Parent.Title
```

Displays the title of the active notebook from the NotebookItems collection.

Path Property

Objects

Read Only
Value: String

Syntax: *Application/Notebook object*.Path

Returns the default path in which SigmaPlot looks for documents, or the path of the specified notebook file.

For notebooks, you can use the Name property to return the file name without the path, or use the FullName property to return the file name and the path together.

Examples

```
MsgBox Path
```

Displays the current SigmaPlot path.

ChDir ActiveDocument.Path

Changes the current directory to the directory of the current notebook file.

Plots Property

Objects

Read Only
Value: Object (Collection)

Syntax: *Graph object*.Plots

Returns the collection of plots for the specified Graph object. Use an index to return the individual Plot objects for the graph.

Example

```
Dim x As Long
x=ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots.Count
Dim SPPlot As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0)
Dim plotobj As Object
Dim plotlist$(ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots.Count)
Dim i
For i=0 To x-1
plotlist$(i)=SPPlot.Plots(i).Name
Next i
Begin Dialog UserDialog 310,133,"Set Current Plot" ' %GRID:10,7,1,1
GroupBox 20,14,160,105,"Available Plots",.GroupBox1
ListBox 30,28,140,84,plotlist(),.ListBox1
OKButton 210,21,80,21
CancelButton 210,56,80,21
End Dialog
Dim dlg As UserDialog
Dialog dlg
Dim index As Long
index=dlg.ListBox1
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(index).SetObjectCurrent
```

Presents the user with a list of plots on the current page. The plot selected from the list is set as the current plot.

Saved Property

Objects

Read Only
Value: Boolean

Syntax: *Notebook/NotebookItems object*.Saved

Returns a True or False value for whether of not the document has been saved since the last changes. Note that notebook items that are closed from within SigmaPlot are automatically saved to the notebook, but that the notebook file is only saved using a Save or Save As command or method.

Example

MsgBox ActiveDocument.Saved

Returns True if the current notebook has had no changes made to it since the last save, or False if the notebook has either never been saved or if changes have been made since the last save.

SelectedText Property

Objects

Read/Write
Value: String

Syntax: *ReportItem object*.SelectedText

Returns the text of the current selection from a ReportItem. You can set or return a text selection using the SelectionExtent property.

Example

```
Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim index2
index2=0
Dim ReportList$(ActiveDocument.NotebookItems.Count)
Dim Item
For Each Item In ActiveDocument.NotebookItems
If ActiveDocument.NotebookItems(Index).ItemType = 5 Then
ReportList$(Index2) = ActiveDocument.NotebookItems(Index).Name
index2=index2+1
End If
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Report Items in Active Notebook" ' %GRID:10,7,1,1

OKButton 210,14,90,21
ListBox 20,14,170,91,ReportList(),.ListBox1
End Dialog
Dim dlg1 As UserDialog
Dialog dlg1
Dim SelectedReport
SelectedReport=dlg1.ListBox1

Begin Dialog UserDialog 400,168,"Insert Text" ' %GRID:10,7,1,1
TextBox 40,35,310,91,.TextBox1,1
Text 40,7,310,21,"Text to insert at beginning of report:",.Text1
OKButton 100,140,90,21
CancelButton 200,140,90,21
End Dialog
Dim dlg2 As UserDialog
Dialog dlg2
Dim RepObj As Object
Set RepObj=ActiveDocument.NotebookItems(ReportList$(SelectedReport))
RepObj.Open
Dim selection(3)
selection(0) = 0
selection(1) = 0
RepObj.SelectionExtent = selection
RepObj.SelectedText=dlg2.TextBox1 + vbCrLf
```

Inserts the entered text at the beginning of a selected report.

SelectionExtent Property

Objects

Read/Write
Value: Variant

Syntax: *ReportItem/ExcelItem object*.SelectionExtent

Returns the array of current selection extents from a ReportItem or ExcelItem. The start and stop indices for each selection are listed as individual members of the array, e.g., .SelectionExtent(0) is the start of the first selection, and SelectionExtent(1) is the end of the first selection.

Example

```
Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim index2
index2=0
Dim ReportList$(ActiveDocument.NotebookItems.Count)
Dim Item
For Each Item In ActiveDocument.NotebookItems
If ActiveDocument.NotebookItems(Index).ItemType = 5 Then
ReportList$(Index2) = ActiveDocument.NotebookItems(Index).Name
index2=index2+1
End If
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Report Items in Active Notebook" ' %GRID:10,7,1,1

OKButton 210,14,90,21
ListBox 20,14,170,91,ReportList(),.ListBox1
End Dialog
Dim dlg1 As UserDialog
Dialog dlg1
Dim SelectedReport
SelectedReport=dlg1.ListBox1

Begin Dialog UserDialog 400,182,"Insert Text" ' %GRID:10,7,1,1
TextBox 30,28,330,70,.TextBox1,1
Text 30,7,340,14,"Text to insert into report:",.Text1
OptionGroup .Group1
OptionButton 50,133,20,14,"OptionButton1",.OptionButton1
OptionButton 50,154,20,14,"OptionButton2",.OptionButton2
Text 50,112,170,14,"Insert at:",.Text2
Text 80,133,140,14,"beginning of report",.Text3
Text 80,154,140,14,"end of report",.Text4
OKButton 300,112,70,21
CancelButton 300,147,70,21
End Dialog
Dim dlg2 As UserDialog
Dialog dlg2

Dim RepObj As Object
Set RepObj=ActiveDocument.NotebookItems(ReportList$(SelectedReport))
RepObj.Open
Dim insertedtext As String
Dim selection(3)
If dlg2.Group1=0 Then
selection(0) = 0
selection(1) = 0
```

```
insertedtext = dlg2.TextBox1 + vbCrLf
Else
selection(0) = -1
selection(1) = -1
insertedtext = vbCrLf + dlg2.TextBox1
End If

RepObj.SelectionExtent = selection
RepObj.SelectedText= insertedtext
```

Inserts the entered text at the beginning or end of the selected report.

```
Begin Dialog UserDialog 280,203,"Define Selection Region" ' %GRID:10,7,1,1
GroupBox 20,7,140,84,"Row Boundaries",.GroupBox1
Text 50,28,50,21,"Top:",.Text1
Text 30,56,70,21,"Bottom:",.Text2
TextBox 90,28,40,21,.TextBox1
TextBox 90,56,40,21,.TextBox2
GroupBox 20,105,140,77,"Column Boundaries",.GroupBox2
Text 40,126,50,14,"Left:",.Text3
Text 40,154,60,14,"Right:",.Text4
TextBox 90,126,40,21,.TextBox3
TextBox 90,154,40,21,.TextBox4
OKButton 190,14,80,21
CancelButton 190,49,80,21
End Dialog
Dim dlg As UserDialog
Dialog dlg
Dim SelectionArray(3)
ActiveDocument.NotebookItems("Excel Worksheet").IsCurrentItem = True
Dim SelectionArray(3)
SelectionArray(0) = CLng(dlg.TextBox3)-1 'left
SelectionArray(1) = CLng(dlg.TextBox1)-1 'top
SelectionArray(2) = CLng(dlg.TextBox4)-1 'right
SelectionArray(3) = CLng(dlg.TextBox2)-1 'bottom
ActiveDocument.CurrentItem.Open ' Bring to top. Must be done to read excel selection

ActiveDocument.CurrentItem.SelectionExtent = SelectionArray
Dim SelectionReturned
SelectionReturned = ActiveDocument.CurrentItem.SelectionExtent
```

Presents a dialog for selecting a region in an Excel worksheet.

ShowStatsWorksheet Property

Objects

Read/Write
Value: Boolean

Syntax: *NativeWorksheetItem object*.ShowStatsWorksheet = *Boolean*

If this Boolean property is set to "True", SigmaPlot opens up a statistics window that displays statistics about the specified NativeWorksheetItem. Statistics include: mean, standard deviation, standard error, half-widths for 95% and 99% confidence intervals, sample size, total, minimum, maximum, smallest positive value, and number of missing values. If this property is set to "False", the statistics window is closed if open.

This property returns "True" if the statistics worksheet window is open or "False" if the worksheet window is not open or the specified NativeWorksheet is not open.

If the specified NativeWorksheet object is not open, setting this property has no effect.

Example

```
ActiveDocument.CurrentDataItem.ShowStatsWorksheet=True
```

Displays column statistics for the current worksheet.

StatsWorksheetDataTable Property

Objects

Read Only
Value: Object

Syntax: *NativeWorksheetItem object*.StatsWorksheetDataTable

Returns the Column Statistics worksheet as a DataTable object.

Returns an object expression representing the read-only data table belonging to the NativeWorksheetItem's statistics worksheet. If the worksheet has not been opened using the ShowStatsWorksheet property, this property returns nothing.

Example

```
Activedocument.CurrentDataItem.ShowStatsWorksheet=True
Dim statsitem As Object
Set statsitem = Activedocument.CurrentDataItem.StatsWorksheetDataTable
Dim statsdata() As Variant
statsdata()=statsitem.GetData(0,0,9,1)
ActiveDocument.NotebookItems("Data 1").DataTable.PutData(statsdata(),0,6)
```

Retrieves the first two rows of the statistics worksheet (the means and standard deviations) for the first 10 columns and places the data in the Data 1 worksheet beginning at row 6.

StatusBar Property

Objects

Read/Write
Value: String

Syntax. StatusBar

Sets or returns the SigmaPlot application window status bar text. Note that when a macro is running within SigmaPlot, it will also issue status messages that will overwrite messages set with the StatusBar property. A macro running in VB or VBA outside SigmaPlot will not create its own status bar messages other than those set with StatusBar.

Examples

```
MsgBox StatusBar
```

Displays the current status bar text.

StatusBar = "My current status"

Sets the status bar to read "My current status."

Subject Property

Objects

Read/Write
Value: String

Syntax: *Notebook/NotebookItems object*.Subject

A standard property of notebook files and all NotebookItems objects. Sets the Subject field under the Summary tab of the Windows 95/98 file Properties dialog box.

Note that the Subject for notebook items is not currently displayed or used.

Examples

ActiveDocument.Subject = "Mammalian Genetics"

Changes the subject of the current notebook to "Mammalian Genetics."

MsgBox Notebooks(0).Subject

Returns and displays the subject used for first open notebook.

SuspendIdle Property

Read/Write
Value: Boolean

Syntax: *Application*.SuspendIdle

Used to allow VisualBasic and other external applications to access some SigmaPlot objects. ***Remember to reset this property to false when finished with the necessary operations***.

Example

This is sample VB code that is used to temporarily suspend SigmaPlot's idle function.

Dim objSPApp As New Application
objSPApp.SuspendIdle = True

Symbols Property

Objects

Read Only
Value: Object

Syntax: *Plot object*.Symbols

Returns the Symbol object for the specified Plot object.

Example

```
Dim SPPlot As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0)
Dim symtype,i As Long
Dim SymbolShape(ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots.Count)

Dim msgtxt As String
For i=0 To SPPlot.Plots.Count-1
symtype=SPPlot.Plots(i).Symbols.GetAttribute(SSA_SHAPE,SymbolShape(i))
msgtxt=msgtxt + SPPlot.Plots(i).Name + ": " + CStr(symtype) + vbCr
Next i
MsgBox msgtxt,"Symbol Shapes"
```

Lists the symbol shape used for each plot on the current page.

Template Property

Objects

Read Only
Value: Object

Syntax. Template

Returns the Notebook object used as the template source file. The template is used for new page creation. To create a graph page using a template file, use the ApplyPageTemplate method.

Example

```
MsgBox (Template.FullName,0+64,Template File)
```

Returns the file name and path for the current template file.

Text Property

Objects

Read/Write
Value: String

Syntax: *ReportItem/TransformItem/MacroItem object*.Text

Specifies the text for the report, transform or macro code. The text is unformatted, plain text.

Note: Use the vbCrLf string data constant to insert a carriage-return and linefeed string.

Transforms: To change the value of a transform variable, use the AddVariableExpression method. Run transforms using the Execute method.

Examples

```
Dim ReportObject As Object
Set ReportObject = ActiveDocument.NotebookItems.Add(CT_REPORT)
ReportObject.Text = "Now is the time for all good men to come to the aid of their parties" + vbCrLf + _

"The quick brown fox jumped over the lazy dog." + vbCrLf + _
"Now is the winter of our discontent."
```

Adds the specified text to a new report item.

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Open
SPTransform.Text = "col(1)=gaussian(1000)" + vbCrLf + "col(2)=histogram(col(1),100)" + vbCrLf

SPTransform.Execute
SPTransform.Name= Path + "\Transforms\My Transform.xfm"
SPTransform.Close(True)
```

Runs a simple transform that generates 1000 normally distributed datapoints and histograms them into 100 bins, then saves it as a file.

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Open
SPTransform.Text = "x=col(1)" + vbCrLf +"erf(x)=1-(.3480242*terf(x)-.0958798*terf(x)^2 + _

.7478556*terf(x)^3)*exp(-x^2)" + vbCrLf +"terf(x)=1/(1+.47047*x)" + vbCrLf + _
"erf1(x)=if(x<0,-erf(-x),erf(x))" + vbCrLf +"P(x)=(erf1(x/sqrt(2))+1)/2" + _
vbCrLf +"col(2)=P(x)*100" + vbCrLf
SPTransform.Execute
SPTransform.Close(False)
```

Computes a Gaussian Cumulative Error Distribution function for column 1 using a transform. Note that all the transform code is placed on a single line, with a + vbCrLf string constant used for line breaks.

TickLabelAttributes Property

Objects

Read Only
Value: Object

Syntax: *Axis object*.TickLabelAttributes

Returns the tick label Text objects for the specified Axis object.

2 Major Tick Labels

3 Minor Tick Labels

Example

```
Dim SPAxisMajor,SPAxisMinor
```

```
Set SPAxisMajor = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(0).TickLabelAttributes(2)

Set SPAxisMinor = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Axes(0).TickLabelAttributes(3)

SPAxisMajor.Color(RGB_BLUE)
SPAxisMajor.SetAttribute(STA_BOLD,True)
SPAxisMajor.SetAttribute(STA_SIZE,140)
SPAxisMinor.Color(RGB_GREEN)
SPAxisMinor.SetAttribute(STA_ITALIC,True)
SPAxisMinor.SetAttribute(STA_SIZE,100)
```

Adjusts the appearance of the tick labels along the x-axis. Major tick labels appear as bold, blue text. Minor tick labels appear as italic, green text. In addition, the minor labels appear smaller than the major labels.

Title Property

Objects

Read/Write
Value: String

Syntax: *Notebook object*.Title

A Notebook object property. Sets the Name of the NotebookItem object of the Notebook file, and the Title field under the Summary tab of the Windows 95/98 file Properties dialog box. Does not affect the file name; to change the file name, use either the Name or FullName property.

Examples

MsgBox Notebooks(0).Title

Returns and displays the entry title used for first open notebook.

ActiveDocument.Title = "Research Project 1 Result"

Changes the entry titleof the current notebook to "Research Project 1 Result."

Top Property

Objects

Read/Write
Value: Long

Syntax: *Notebook/NotebookItems document object*.Top

Sets or returns the top coordinate of the application window or specified notebook document window.

Examples

ActiveDocument.NotebookItems("Data 1").Top = 0

Sets the top of the "Data 1" notebook item window to 0.

MsgBox Top

Displays the top coordinate of the SigmaPlot application window.

Note: To set the window size and position simultaneously, use a With statement:

```
With Application

.Left = 0
.Top = 0
.Height = 600
.Width = 800
End With
```

TrigUnit Property

Objects

Read/Write
Value: Integer

Syntax: *TransformItem/FitItem/PlotEquation object*.TrigUnit

Sets the angular unit for arguments in trigonometric functions as it is passed to the evaluator. This overrides any setting that may be contained in a transform file.

This does not read or set the trig units set for any given file, but only the default trig units used by the transform engine.

Trig Unit	Value
Radians	0
Degrees	1
Grads	2

Example

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Name = "d:\Program Files\SigmaPlot\SPW6\My Transform.xfm"

SPTransform.Open
SPTransform.TrigUnit = 0
SPTransform.Execute
SPTransform.Close(False)
```

Opens the transform file "My Transform.xfm" and runs it using radians as the trig units.

UpperPickIndex Property

Objects

Read Only
Value: Long

Syntax: *GraphWizard object*.UpperPickIndex

Returns the upper range of the index(s) picked by the graph wizard. See also LowerPickIndex.

The upper index is the last column picked to plot for the graph created by the finishing of the GraphWizard object. These values are not correctly initialized until the graph wizard has run to completion.

The upper and lower indexes correspond to the indexes data titles set by the SetTitles method.

See the GraphWizard object for examples of using the upper and lower index values.

Visible Property

Objects

Read/Write
Value: Boolean

Syntax: *Application/Notebook/NotebookItems document object*.Visible

A property common to the Application, Notebook, and NotebookItems document objects. Sets or returns a Boolean indicating whether or not the application or specified document window is visible. Do not set the Application property to False from within SigmaPlot or you will lose access to the application.

Note that hidden document windows will still appear in the notebook window tree. Setting Visible=False for a notebook object hides all document windows for the notebook as well.

Examples

```
ActiveDocument.Visible=False
```

Hides the current notebook and all windows for that notebook. This is useful if you need to use a "hidden" worksheet to perform computations.

```
Dim SPApp As Object
Set SPApp = CreateObject("SigmaPlot.Application.1")

Visible=False
```

Creates a SigmaPlot application object from VB or VBA, and makes the SigmaPlot window hidden. Note that when SigmaPlot is launched from another application (such as VB or VBA) the default condition is Visible=False.

Width Property

Objects

Read/Write

Value: Long

Syntax: *Notebook/NotebookItems document object*.Width

Sets or returns the width of the application window or specified notebook document window.

Examples

```
ActiveDocument.NotebookItems("Data 1").Width = 600
```

Sets the width of the "Data 1" notebook item window to 600.

MsgBox Width

Displays the width of the SigmaPlot application window.

Note: To set the window size and position simultaneously, use a With statement:

```
With Application

.Left = 0
.Top = 0
.Height = 600
.Width = 800
End With
```

SigmaPlot Methods

For Fit Item or FitResult Properties, see FitItem and FitResults Properties and Methods

About Methods

Activate

Add

AddVariableExpression

AddWizardAxis

AddWizardPlot

ApplyPageTemplate

Clear

Close

ColumnBorderThickness

Copy

CreateGraphFromTemplate

CreateWizardGraph

Cut

Delete

DeleteCells

Execute

Export

GetAttribute

GetData

GetMaxLegalSize

GetMaxUsedSize

Goto

Help

Import

InsertCells

Interpolate3DMesh

IsRegionWriteProtected

Item

LaunchWizard

ModifyWizardPlot

NormalizeTernaryData

Open

Paste

Print

PrintStatsWorksheet

PutData

Redo

Remove

Run

RunEditor

Quit

Save

SaveAs

Select

SelectAll

SelectObject

SetAttribute

SetCurrentObjectAttribute

SetObjectCurrent

Activate Method

Objects

Type: Sub

Syntax: *Notebook object*.Activate

Makes the specified notebook the object specified by the ActiveDocument property.

Example

```
Notebooks("c:\SigmaPlot\My Notebook.jnb").Activate

MsgBox ActiveDocument.Title
```

Makes the specified notebook the active document, then displays the notebook title.

Add Method

Objects

Type: Function
Result: Object

Syntax: *collection*.Add(*parameters*)

The Add method is used in collections to add a new item to the collection. The parameters depend on the collection type:

Collection	**Value**	**Parameters**	**Object Added**
Notebooks		None	Notebook
NotebookItems	1.	CT_WORKSHEET	NativeWorksheetItem
	2.	CT_GRAPHICPAGE	GraphItem
	3.	CT_FOLDER	SectionItem
	4.	CT_STATTEST	ReportItem (SigmaStat)
	5.	CT_REPORT	ReportItem (SigmaPlot)

	6.	CT_FIT	FitItem
	7.	CT_NOTEBOOK	NotebookItem
	8.	CT_EXCELWORKSHEET	ExcelItem
	9.	CT_TRANSFORM	TransformItem
	1.		MacroItem
GraphObjects:	2	GPT_GRAPH, more...	Graph
	3	GPT_PLOT, more...	Plot
	4	GPT_AXIS, more...	Axis
	5	GPT_TEXT, more...	Text
	6	GPT_LINE, more...	Line
	7	GPT_SYMBOL, more...	Symbol
	8	GPT_SOLID, more...	Solid
	9	GPT_TUPLE, more...	Tuple
	10	GPT_FUNCTION, more...	Function
	1.	GPT_EXTERNAL, more...	GraphObject
	1.	GPT_BAG, more...	Group
NamedRanges		*Name string, Left long, Top long, Width long, Height long*	NamedRange

The GraphObjects collection uses the CreateGraphFromTemplate and CreateWizardGraph methods to create new GraphObject objects.

Examples

Notebooks.Add

Creates a new notebook.

ActiveDocument.Add(8)

Adds an in-place activated Excel worksheet to the current notebook, at the position of the current notebook item.

```
Dim Group_A As String
Group_A = "Group A"
ActiveDocument.NotebookItems("Data 1").DataTable.NamedRanges.Add(Group_A,0,0,1,-1)
```

Adds the column title "Group A" to column 1 of the "Data 1" worksheet.

Adding Graphs

The following example demonstrates the addition of graphs to a page and the addition of plots and "tuples" to a graph.

```
Dim ANotebook As Object
Set ANotebook = Notebooks.Add
Dim DataItem As Object
Set DataItem = ANotebook.NotebookItems("Data 1")
Dim ADataTable As Object
Set ADataTable = DataItem.DataTable

'Create some example data.
Dim i
For i = 1 To 5
ADataTable.Cell(0,i-1) = i
ADataTable.Cell(1,i-1) = i+1
ADataTable.Cell(2,i-1) = i+2
ADataTable.Cell(3,i-1) = i+3
ADataTable.Cell(4,i-1) = i+4
```

```
Next i
Dim Sign
Sign = 1
For i = 1 To 5
ADataTable.Cell(5,i - 1) = 100 + i*Sign
Sign = -Sign
Next i

'Create graphics page in the notebook
Dim GraphicPage
Set GraphicPage = ANotebook.NotebookItems.Add(CT_GRAPHICPAGE)

'Create a graph manually. (This isn't recommended. Better to use CreateWizardGraph)
Dim PageObject As Object
Set PageObject = GraphicPage.GraphPages(0)

Dim AGraphObject As Object
Set AGraphObject = PageObject.ChildObjects.Add(GPT_GRAPH, SGA_COORD_CART2, SLA_TYPE_BAR, SLA_SUBTYPE_VERTY)


MsgBox("Count of plots in graph: " + CStr(AGraphObject.Plots.Count),0+64,"Plot Count")

Dim PlotObject As Object
Set PlotObject = AGraphObject.Plots(0)

'Plot objects only allow you to add objects of type GPT_TUPLE
'Add 4 tuples to make a grouped bar chart with groups of 4.
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,1)
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,2)
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,3)
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,4)

MsgBox("Count of tuples in plot: " + CStr(PlotObject.ChildObjects.Count),0+64,"Tuple Count")

' Get some repeat type schemes for the two tuples.
Dim FillScheme
FillScheme = PlotObject.StockScheme(STOCKSCHEME_PATTERN_OLDINCREMENT)
' Tell the plot to use the "old increment" scheme"
PlotObject.Fill.SetAttribute(SDA_PATTERNREPEAT, FillScheme)
' Set the initial density and pattern
PlotObject.Fill.SetAttribute(SDA_PATTERN, (SDA_DENS_FINE*&H10000) + SDA_PAT_HOLLOW)

'Get some repeat type schemes for the two tuples.
Dim ColorScheme
ColorScheme = PlotObject.StockScheme(STOCKSCHEME_COLOR_GRAYS)

'Tell the plot to use the "gray" scheme"
PlotObject.Fill.SetAttribute(SDA_COLORREPEAT, ColorScheme)
' Set the initial color in the pattern
PlotObject.Fill.SetAttribute(SDA_COLOR, RGB_GRAY)

'Add a line plot to the graph.
Set PlotObject = AGraphObject.Plots.Add(GPT_PLOT, SLA_TYPE_SCATTER, SLA_SUBTYPE_NORMAL)
'Plot objects only allow you to add objects of type GPT_TUPLE
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,5)

'Turn on the line for the scatter plot
PlotObject.SetAttribute(SLA_PLOTOPTIONS, FlagOn(SLA_FLAG_LINEON))
'Make it a spline.
PlotObject.SetAttribute(SLA_LINEPATH, SLA_PATH_SPLINE)

'Set the main plot line's attributes. Make sure it is selected
'by deselecting all drop lines and function lines.
PlotObject.SetAttribute(SLA_SELECTFUNC,SLA_FUNC_NONE)
PlotObject.SetAttribute(SLA_SELECTDROP,DIM_NONE)
'Set the main line color
PlotObject.SetAttribute(SEA_COLOR,RGB_RED)
'Make sure the graph and plot are current
AGraphObject.SetObjectCurrent
PlotObject.SetObjectCurrent
'Add a new Y axis
GraphicPage.AddWizardAxis(SAA_TYPE_LINEAR,DIM_Y,AxisPosRightNormal)
```

Adding Drawing Objects

```
'Create a normal line
Dim Points()
Redim Points(3)
Points(0) = -3520
Points(1) = 2479
Points(2) = -2187
Points(3) = 3188
Dim LineObject As Object
Set LineObject = ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Add(GPT_LINE, Points)
LineObject.SetAttribute(SEA_END2TYPE, 1) ' normal line end
'Create an arrow
Redim Points(3)
Points(0) = -687
Points(1) = 3167
Points(2) = 21
Points(3) = 1896
Set LineObject = ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Add(GPT_LINE, Points)
LineObject.SetAttribute(SEA_END2TYPE, 2) ' arrow line end

'Create a box
Redim Points(3)
Points(0) = -3041
Points(1) = 896
Points(2) = -375
Points(3) = -250
ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Add(GPT_SOLID, Points, SOA_EXT_RECT)

'Create an ellipse
Redim Points(3)
Points(0) = 0
Points(1) = 833
Points(2) = 2146
Points(3) = -333
ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Add(GPT_SOLID, Points, SOA_EXT_ELLIPSE)


'Select all objects
ActiveDocument.CurrentPageItem.Select(False, -4854, 3625, 2937, -2812)

'Make them red.
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SOA_COLOR, &H000000ff)
```

Adds red drawing objects to the graph page.

Adding Text

```
Dim Points()
Redim Points(1)
Points(0) = 2041
Points(1) = 1958
ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Add(GPT_TEXT, "", Points)
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETOBJECTATTR, STA_ORIENTATION, 0)
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETOBJECTATTR, STA_RTF , _
"{\rtf1\ansi0{\colortbl\red0\green0\blue0;}\deff0{\fonttbl\f0\fnil Arial;}\ql\sl200\slmult0\f0\cf0\up0\fs20\i0\b0\ul0Outlier}")
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETOBJECTATTR, STA_OPTIONS, &H00008001)
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETOBJECTATTR, STA_PARAGRAPHJUSTIFY,
STA_JUSTIFY_LEFT)
```

Adds the term "Outlier" at the specified location in the current graph.

AddVariableExpression Method

Objects

Type: Sub

Syntax: *TransformItem object*.AddVariableExpression(*variable name* string, *variable value* variant)

Allows the substitution of any transform variable with a value.

Examples

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Open
SPTransform.Text = "col(c)=histogram(col(r),b)" + vbCrLf
Dim HistogramParameters(2)
HistogramParameters(0) = "1"
HistogramParameters(1)= "col(2)"
HistogramParameters(2)= "3"
SPTransform.AddVariableExpression("r", HistogramParameters(0))

SPTransform.AddVariableExpression("b", HistogramParameters(1))

SPTransform.AddVariableExpression("c", HistogramParameters(2))

SPTransform.Execute
SPTransform.Close(False)
```

Declares and uses a HistogramParameters array as the parameter values for the histogram transform function. The following macro uses values returned from a dialog to provide the parameters for the gaussian transform function:

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Open
SPTransform.Text = "col(c)=gaussian(n,0/0,m,s)" + vbCrLf
Begin Dialog
UserDialog 320,126,"Normally Distributed Numbers" ' %GRID:10,7,1,1

OKButton 210,7,90,21
CancelButton 210,35,90,21
TextBox 100,7,90,21,.n
TextBox 100,35,90,21,.mean
TextBox 100,63,90,21,.stddev
TextBox 100,91,90,21,.Results
Text 10,10,80,14,"Number",.Text1
Text 10,38,60,14,"Mean",.Text2
Text 10,66,90,14,"Std Dev",.Text4
Text 10,94,80,14,"Results Col",.Text3
End Dialog
Dim dlg As UserDialog
dlg.n = "100"
dlg.mean = "1"
dlg.stddev = ".25"
dlg.Results = "1"
Dialog dlg
SPTransform.AddVariableExpression("n", dlg.n)
SPTransform.AddVariableExpression("m", dlg.mean)
SPTransform.AddVariableExpression("s", dlg.stddev)
SPTransform.AddVariableExpression("c", dlg.Results)
SPTransform.Execute
SPTransform.Close(False)
```

AddWizardAxis Method

Objects

Type: Sub

Syntax: *GraphItem object*. AddWizardAxis (*scale type*, *optionaldimension, optional position*)

Adds an additional axis to the current graph and plot on the specified GraphItem object, using the AddWizardAxis options. If there is only one plot for the current graph, SigmaPlot will return an error. Use the following parameters to specify the type of scale, the dimension, and the position for the new axis:

ScaleType

SAA_TYPE_LINEAR

SAA_TYPE_COMMON (Base 10)

SAA_TYPE_LOG (Base e)

SAA_TYPE_PROBABILITY

SAA_TYPE_PROBIT

SAA_TYPE_LOGIT

Dimension

DIM_X	1.	The X dimension
DIM_Y	2.	The Y dimension
DIM_Z	3.	The Z dimension (if applicable)

Position

AxisPosRightNormal	0
AxisPosRightOffset	1.
AxisPosTopNormal	2.
AxisPosTopOffset	3.
AxisPosLeftNormal	4.
AxisPosLeftOffset	5.
AxisPosBottomNormal	6.
AxisPosBottomOffset	7.

Example

```
Dim GraphPage As Object
Set GraphPage = ActiveDocument.CurrentPageItem
Dim ColumnList(0)
ColumnList(0) = 1
GraphPage.AddWizardPlot("Scatter Plot", "Simple Scatter", "Single Y", ColumnList)

GraphPage.AddWizardAxis(SAA_TYPE_COMMON,2,0)
```

Adds a scatterplot to the current plot. The Y-axis for the scatterplot employs a lagarithmic scale and is positioned along the right border of the plot.

AddWizardPlot Method

Objects

Type: Function
Results: Boolean

Syntax: *GraphItem object*.AddWizardPlot(*required parameters variants, optional parameters variants*)

Adds another plot to the current graph on the specified GraphItem object using the following parameters to define the plot:

Parameter	**Values**	**Optional**
graph type	any valid type name	no
graph style	any valid style name	no
data format	any valid data format name	no
column array	any column number/title array	no
columns per plot array	array of columns in each plot	yes
error bar source	any valid source name	error bar plots only
error bar computation	any valid computation name	error bar plots only
anglular axis units	any valid angle unit name	polar plots only
lower range bound	any valid degree value	polar plots only
upper range bound	any valid degree value	polar plots only
ternary units	upper range of ternary axis scale	ternary plots only
lower error bar computation	any valid computation name	error bar plots only
row selection	Boolean: True allows selection of a row range for y-replicate (row-summary) plots. Use False to support pre-y replicate data format macros.	Row summary plots only

Example

```
Dim GraphPage As Object
Set GraphPage = ActiveDocument.NotebookItems.Add(CT_GRAPHICPAGE)
Dim ColumnList(0)
ColumnList(0) = 0
GraphPage.CreateWizardGraph("Vertical Bar Chart", "Simple Bar", "Single Y", ColumnList)

ColumnList(0) = 1
GraphPage.AddWizardPlot("Scatter Plot", "Simple Scatter", "Single Y", ColumnList)
```

Adds a simple scatter plot of the data in the second column to a vertical bar chart of the data in the first column.

ApplyLayoutTemplate Method

Type: Sub

Syntax: *GraphItem object*.ApplyLayoutTemplate(*template name variant, optional template file name variant*)

Applies a page layout (graph arrangement) to the specified graph page. The graph page must be open.

Example

```
Dim SourceTemplate As String
SourceTemplate = "2 up, 3"" x 3""" 'To use a quote (") in a string, use two quotes ("")

Dim SourceFile As String
ActiveDocument.CurrentPageItem.ApplyLayoutTemplate(SourceTemplate)
```

Applies the 2 up 3" x 3" graph layout from the default layout notebook to the current page.

ApplyPageTemplate Method

Objects

Type: Function
Results: Boolean

Syntax: *GraphItem object*.ApplyPageTemplate(*template name string*, *optional template file name string*)

Overwrites the current GraphItem using a new page template specified by the template name. Optionally, you can specify the notebook file to use as the source of the template page. If no template file is specified, the default template notebook is used, as returned by the Template property.

Examples

```
Dim TemplatePage As String
TemplatePage = "Scatter Plot"
ActiveDocument.CurrentPageItem.ApplyPageTemplate(TemplatePage)
```

Applies the "Scatter Plot" template page from the default SigmaPlot template notebook to the current page.

```
Dim SourceTemplate As String
SourceTemplate = "Graph Page 1"
Dim SourceFile As String
SourceFile = "d:\My Documents\Old Notebook.jnb"
ActiveDocument.CurrentPageItem.ApplyPageTemplate(SourceTemplate,SourceFile)
```

Applies the "Graph Page 1" page from d:\My Documents\Old Notebook.jnb as the template for "Graph Page 2" in My Notebook.jnb.

BoldFont Method

Objects

Type: Sub

Syntax: *ReportItem object*.BoldFont

Toggles the bold font effect for the selected text.

See the ReportItem object for an example of selection and formatting.

ChangeDefaultFont Method

Objects

Type: Sub

Syntax: *ReportItem object*.ChangeDefaultFont

Specifies the font name for the report.

See the ReportItem object for an example of selection and formatting.

Clear Method

Objects

Type: Sub

Syntax: *NotebookItems object*.Clear

Clears the selection in items that support this.

Examples

```
ActiveDocument.CurrentDataItem.Clear
```

Clears the selected cells in the current worksheet.

```
ActiveDocument.CurrentItem.Clear
```

Clears the currently selected item.

Close Method

Objects

Type: Sub

Syntax: *object*.Close(*save parameters*)

The Close method is used to close notebooks and notebook items. The parameters for each object type depend on the object:

Notebook	*Save before closing boolean, filename string*
NotebookItems *Save before closing boolean*	

Specifying a *Save before closing* value of "False" closes the notebook or notebook item without saving changes made to the object.

Note that for NotebookItems and SectionItems, a Close corresponds to an Expanded = False.

Examples

```
Dim FileName As String
FileName = "My Notebook.jnb"
Notebooks(0).Close(True,FileName)
```

Closes the first notebook, saving first to the file name My Notebook.jnb. Note that when no path is specified, the DefaultPath is used.

```
ActiveDocument.NotebookItems("Data 1").Close(False)
```

Closes the Data 1 worksheet window without saving any changes made since the worksheet was first opened.

ColumnTitle Method

Objects

Type: Sub
Result: Variant

Syntax: *DataTable object.*ColumnTitle(*column long,Title* variant)

Gets or sets the column title for the specified column number for the specified data table.

Copy Method

Objects

Type: Sub

Syntax: *NotebookItems object*.Copy

Copies the currently selected item within the specified notebook item. If no item is selected, then an error is returned.

Example

```
Dim MaxColumn As Long
Dim MaxRows As Long
MaxColumn = 0
MaxRows = 0
ActiveDocument.CurrentDataItem.DataTable.GetMaxUsedSize(MaxColumn,MaxRows)
Dim collist$()
ReDim collist$(MaxColumn+1)
Dim i
For i=1 To MaxColumn
collist$(i)=CStr(i)
Next i
Dim msgtext
Begin Dialog UserDialog 400,98,"Copy Column" ' %GRID:10,7,1,1
ComboBox 150,49,50,42,collist(),.ComboBox1
Text 30,14,240,21,"Current Worksheet: " + ActiveDocument.CurrentDataItem.Name,.Text1

Text 30,49,110,21,"Column to copy:",.Text2
OKButton 300,14,80,21
End Dialog
Dim dlg1 As UserDialog
Do
dlg1.ComboBox1="1"
Dialog dlg1
If CLng(dlg1.ComboBox1)>MaxColumn Or CLng(dlg1.ComboBox1)<1 Then
msgtext="Value must be between 1 and " + CStr(MaxColumn)
MsgBox(msgtext,0+48,"Out of Range")
End If
Loop Until CLng(dlg1.ComboBox1)>0 And CLng(dlg1.ComboBox1)<MaxColumn+1

Dim Selection(3)
Selection(0) = CLng(dlg1.ComboBox1)-1
Selection(1) = 0
Selection(2) = CLng(dlg1.ComboBox1)-1
Selection(3) = &H7FFFFFF
ActiveDocument.CurrentDataItem.SelectionExtent = Selection
ActiveDocument.CurrentDataItem.Copy
```

Copies the selected column from the current worksheet to the clipboard.

CreateGraphFromTemplate Method

Objects

Type: Function
Results: Boolean

Syntax: *GraphItem object*.CreateGraphFromTemplate(*graph type variant*, *graph style variant*)

Create a graph for a GraphItem from the Graph Style Gallery. Not yet implemented as a feature.

CreateSmoother Method

Objects

Type: Function
Results: Object

Syntax: *NativeWorksheet/ExcelItem object*.CreateSmoother

Creates a Smoothers object for the specified worksheet item.

Example

```
Dim SPSmoother As Object
Set SPSmoother = ActiveDocument.CurrentDataItem.CreateSmoother
```

CreateWizardGraph Method

Objects

Type: Function
Results: Boolean

Syntax: *GraphItem object*.CreateWizardGraph(*required parameters variants, optional parameters variants*)

Creates a graph in the specified GraphItem object using the Graph Wizard options. These options are expressed using the following parameters:

Parameter	**Values**	**Optional**
graph type	any valid type name	no
graph style	any valid style name	no
data format	any valid data format name	no

columns plotted	any column number/title array	no
columns per plot	array of columns in each plot	yes
error bar source	any valid source name	error bar plots only
upper error bar computation	any valid computation name	error bar plots only
anglular axis units	any valid angle unit name	polar plots only
lower range bound	any valid degree value	polar plots only
upper range bound	any valid degree value	polar plots only
ternary units	upper range of ternary axis scale	ternary plots only
lower error bar computation	any valid computation name	error bar plots only
row selection	Boolean: True allows selection of a row range for y-replicate (row-summary) plots. Use False to support pre-y replicate data format macros.	Row summary plots only

Examples

```
ActiveDocument.NotebookItems.Add(2) 'Adds a new graph page
Dim PlottedColumns(1) As Variant
PlottedColumns(0) = 0
PlottedColumns(1) = 1
ActiveDocument.NotebookItems("Graph Page 1").CreateWizardGraph("Vertical Bar Chart", _

"Simple Bar","XY Pair",PlottedColumns)
```

Plots columns 1 and 2 as a simple bar chart

```
Dim GraphPage As Object
Set GraphPage = ActiveDocument.NotebookItems.Add(CT_GRAPHICPAGE) 'Adds a new graph page

Dim PlottedColumns(9) As Variant
PlottedColumns(0) = 0
PlottedColumns(1) = 1
PlottedColumns(2) = 2
PlottedColumns(3) = 3
PlottedColumns(4) = 4
PlottedColumns(5) = 6
PlottedColumns(6) = 7
PlottedColumns(7) = 8
PlottedColumns(8) = 9
PlottedColumns(9) = 10
Dim ColumnsPerPlot(1) As Variant
ColumnsPerPlot(0) = 5
ColumnsPerPlot(1) = 5 'remaining columns are automatically plotted
GraphPage.CreateWizardGraph("Scatter Plot", _
"Multiple Error Bars & Regression","X Many Y",PlottedColumns,ColumnsPerPlot, _
"Column Means","Standard Deviation")
```

Plots columns 1-5 and 7-11 as column averaged scatter plots with error bars and regression lines.

Cut Method

Objects

Type: Sub

Syntax: *object*.Cut

Removes the current selection from the specified object, placing the contents on the clipboard. This method is equivalent to using the Copy method, followed by the Clear method. However, whereas Copy places OLE link formats on the clipboard for GraphItem objects, Cut does not.

Example

```
ActiveDocument.NotebookItems("Graph Page 1").Cut
```

Cuts the selected objects on "Graph Page 1" to the clipboard.

Delete Method

Objects

Type: Sub

Syntax: *NotebookItems collection*.Delete(*index*)

Deletes a notebook item from a NotebookItems collection, as specified using an index number or name. If the item does not exist, an error is returned.

Example

```
ActiveDocument.NotebookItems.Delete("Data 3")
```

Removes the "Data 3" notebook item from the notebook.

DeleteCells Method

Objects

Type: Function
Results: Boolean

Syntax: *NativeWorksheetItem.*DeleteCells(*left long, top long, right long, bottom long, direction long*)

Deletes the specified cells from the worksheet. The remaining cels can be moved in two different directions to fill in the deleted region:

1. Shift Cells Up

2. Shift Cells Left

To delete an entire column or row, simply set the column bottom or row right value to the system maximum:

Rows: 32,000,000
Columns: 32,000

Examples

```
ActiveDocument.NotebookItems(2).DeleteCells(0,0,0,99,2)
```

Deletes the block column 1, row 1 to column 1, row 100 and shifts the adjacent data to the left.

```
ActiveDocument.NotebookItems("Data 1").DeleteCells(0,4,32000,4,1)
```

Deletes row 5 and shifts the rows below up one.

Execute Method

Objects

Type: Sub

Syntax: *TransformItem object*.Execute

Used to execute the specified TransformItem.

Example

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)

SPTransform.Name = Path + "\Transforms\Mesh.xfm"
SPTransform.Open
SPTransform.Execute
SPTransform.Close(False)
```

Opens the example transform Mesh.xfm using the application path, then executes it.

Export Method

Objects

Type: Sub

Syntax: *object*.Export(*FileName* variant, *FormatName* variant)

Exports the specified notebook item to a new file. SigmaPlot supports export of NativeWorksheetItem, GraphItem, ReportItem, and NotebookItem objects.

- If applied to a NativeWorksheetItem object, this method exports either the data in the worksheet to the specified data format or the entire notebook to a previous SPW file format.
- If applied to a GraphItem object, this method exports either the graphic data on the page to the specified graphic format or the entire notebook to a previous SPW file format.
- If applied to the first NotebookItem in the NotebookItemList, this method exports the entire notebook to a previous SPW file format.

The Export method supports the following formats:

Data file formats (for NativeWorksheet objects):

FormatName	Data File Type
XLS4	Excel 4
XLS3	Excel 3

CSV	Comma Delimited ASCII
TAB	Tab Delimited ASCII
TXT	Plain Text ASCII
DIF	DIF
WKS	Lotus 1-2-3 v1.0
DB2	DBase II
DB3	DBase III
WQ1	Quattro Pro v1.0
DB	Paradox v3.0
WRK	Symphony v1.0
SYS	SYSTAT

Graphic file formats (for GraphItem objects):

FormatName	Data File Type
SPW	SigmaPlot 2.0, 1.0,
BMP	Bitmap
TIF	TIFF
WMF	Metafile
EPS	Encapsulated PostScript
JPG	JPEG

Text file formats (for ReportItem objects):

FormatName	Data File Type
RTF	Rich Text Format
TXT	Plain text
HTM	HTML

Previous version file formats (for Notebook objects):

FormatName	Data File Type
JNB3	SigmaPlot 3.0. SigmaStat 2.0
JNB4	SigmaPlot 4.0, SigmaStat 2.01
SPW	SigmaPlot 2.0, 1.0, SigmaPlot Mac 5.0 data, SigmaScan, SigmaScan Pro, Mocha

Examples

```
GraphPage.Select(-5500,5500,5500,-5500,False)

GraphPage.Export("c:\MyGraph.JPG","JPG")
```

Exports the current graph as a JPG file.

```
ActiveDocument.NotebookItems("Data 1").Export("c:\TestXLS.XLS","XLS4")
```

Exports the "Data 1" worksheet as an Excel (version 4.0) file.

```
ActiveDocument.NotebookItems("Mybook").Export("c:\testJnb3.jnb","jnb3")
```

Exports the MyBook notebook as a SigmaPlot 3 file.

GetAttribute Method

Objects

Type: Function
Result: Long

Syntax: *Page object/child object.*GetAttribute(*attribute, parameter*)

The GetAttribute method is used by all graph page objects to retrieve current attribute settings. Attributes are numeric values that also have constants assigned to them. For a list of all these attributes and constants, see **SigmaPlot Constants**.

Message Forwarding: If you use the GetAttribute method to retrieve an attribute that does not exisit for the current object, the message is automatically routed to an object that has this attribute using the message forwarding table.

Using the Object Browser to view Constants You can view alternate values for attributes and constants by selecting the current attribute value, then clicking the Object Browser button. All valid alternate values will be listed—to use a different value, select the value and click Paste.

Example

```
Dim x As Long
x=ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots.Count
Dim SPPlot As Object
Set SPPlot = ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0)
Dim plotobj As Object
Dim plotlist$(ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots.Count)

Dim i
For i=0 To x-1
plotlist$(i)=SPPlot.Plots(i).Name
Next i
Begin Dialog UserDialog 310,133,"Available Plots" ' %GRID:10,7,1,1
ListBox 30,28,140,84,plotlist(),.ListBox1
OKButton 210,21,80,21
CancelButton 210,56,80,21
End Dialog
Dim dlg1 As UserDialog
Dialog dlg1
Dim index As Long
index=dlg1.ListBox1
Set SPPlot=ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(index)
Dim SymbolShape,SymbolColor,SymbolSize,LineType,LineShape,LineColor
Dim sshape,scolor,ssize,ltype,lshape,lcolor
sshape=SPPlot.Symbols.GetAttribute(SSA_SHAPE,SymbolShape)
scolor=SPPlot.Symbols.GetAttribute(SSA_COLOR,SymbolColor)
ssize=SPPlot.Symbols.GetAttribute(SSA_SIZE,SymbolSize)
ltype=SPPlot.Line.GetAttribute(SEA_LINETYPE ,LineType)
lcolor=SPPlot.Line.GetAttribute(SEA_COLOR ,LineColor)
lshape=SPPlot.Line.GetAttribute(SLA_LINEPATH ,LineShape)
Begin Dialog UserDialog 360,175,"Plot Summary" ' %GRID:10,7,1,1
GroupBox 20,14,150,105,"Symbols",.GroupBox1
Text 30,35,130,14,"Size: "+CStr(ssize),.Text1
Text 30,63,130,14,"Shape: "+CStr(sshape),.Text2
Text 30,91,130,14,"Color: "+CStr(scolor),.Text3
```

```
GroupBox 190,14,150,105,"Lines",.GroupBox2
OKButton 130,133,100,28
Text 200,35,130,14,"Type: "+CStr(ltype),.Text4
Text 200,63,130,14,"Shape: "+CStr(lshape),.Text5
Text 200,91,130,14,"Color: "+CStr(lcolor),.Text6
End Dialog
Dim dlg As UserDialog
Dialog dlg
```

Displays the symbol and line characteristics for the selected plot.

GetData Method

Objects

Type: Function
Result: Variant

Syntax: *DataTable object.*GetData(*left long, top long, right long, bottom long*)

Returns the data within the specified range from a DataTable object as a variant. To ensure that GetData retrieves all data in a row or column, specify the worksheet maximum as the right of bottom parameter.

Examples

```
ActiveDocument.NotebookItems("Data 2").DataTable.GetData(0,99,32000,0)
```

Retrieves all data from row 100.

```
Dim SPData() As Variant
SPData() = ActiveDocument.NotebookItems("Data 1").DataTable.GetData(0,0,1,3)

ActiveDocument.NotebookItems("Data 1").DataTable.PutData(SPData,3,0)
```

Retrieves the data block from (1, 1) to (2,4) and places it as a block starting in column 4.

GetMaxLegalSize Method

Objects

Type: Sub

Syntax: *DataTable object.*GetMaxLegalSize(*maximum columns long, maximum rows long*)

Initializes the values of the maximum worksheet column and row values, so that they can be returned as a variables.

Example

```
Dim MaxColumn As Long
Dim MaxRows As Long
MaxColumn = 0
MaxRows = 0
ActiveDocument.NotebookItems("Data 1").DataTable.GetMaxLegalSize(MaxColumn,MaxRows)

MsgBox CStr(MaxColumn) + ", "+ CStr(MaxRows)
```

Displays the maximum column and row numbers.

GetMaxUsedSize Method

Objects

Type: Sub

Syntax: *DataTable object.*GetMaxUsedSize(*maximum used columns long, maximum used rows long*)

Initializes the values of the maximum used worksheet column and row values, so that they can be returned as a variables.

Example

```
Dim MaxColumn As Long
Dim MaxRows As Long
MaxColumn = 0
MaxRows = 0
MsgBox ActiveDocument.NotebookItems("Data 1").DataTable.GetMaxUsedSize(MaxColumn,MaxRows)

MsgBox CStr(MaxColumn) + ", "+ CStr(MaxRows)
```

Displays the column and row numbers for the last datapoint in the worksheet.

GetPickRange Method

Objects

Type: Function
Result: Boolean

Syntax: *GraphWizard object.*GetPickRange(*lower long, upper long*)

This method returns the ranges set for the picked columns of a GraphWizard object. These have to be previously defined by running the GraphWizard object to completion.

Goto Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem/ExcelItem object.*Goto(*row long, column long*)

Moves worksheet cursor position to the specified cell coordinate for the current NativeWorksheetItem or ExcelItem object.

Example

```
ActiveDocument.NotebookItems("Data 1").Goto(49999,999)
```

Moves the current worksheet cell to row 50,000, column 1000.

Help Method

Objects

Type: Sub

Syntax: Help(*filename variant, ID variant, Index variant*)

Opens an on-line Windows help file to a specific topic context map ID number (as a long) or search index keyword (K-word). You can use either the ID number or an index keyword. If any of the parameters are left empty, the SigmaPlot help file defaults are used.

Examples

```
Dim HelpID As Variant

HelpID = 20
Help(,HelpID)
```

Opens the help topic on the Column tab of the Column and Row Titles Dialog found in the SPW5 help file.

```
Dim ObjectHelp, HelpID As Variant
ObjectHelp = Path + "\SigmaPlot Automation.hlp"

HelpID = 99
Help(ObjectHelp,HelpID)
```

Opens the "Help Method" topic found in the "Sigmaplot Automation.hlp" help file.

Import Method

Objects

Type: Function
Results: Boolean

Syntax: *NativeWorksheetItem.*Import(*file name string, destination column variant, destination row variant, source left variant, source top variant, source right variant, source bottom variant, optional extension string, optional sheet number integer*)

Imports a data file with the specified file name into an existing NativeWorksheetItem. You can specify both the import starting location in the SigmaPlot worksheet, as well as the range of data imported.

Note that you must specify the data file name extension, as the SigmaPlot import filters recognize file types by extension. SigmaPlot can import the following file types:

Tabbed text

Comma delimited text

Excel
Lotus 1-2-3

SigmaPlot/SigmaStat

Systat

TableCurve
SigmaScan

Quattro
FoxPro
Access

Example

```
Dim FileName As String
FileName = "c:\My Documents\Book1.xls"
ActiveDocument.NotebookItems("Data 1").Import(FileName,0,0,0,0,9,255,"xls",1)
```

Imports rows 1 through 256 from columns 1 through 10 in the "Book1.xls" file starting at row 1, column 1, sheet 1.

InsertCells Method

Objects

Type: Function
Results: Boolean

Syntax: *NativeWorksheetItem.*InsertCells(*left long, top long, right long, bottom long, direction long*)

Inserts the specified block of cells into the worksheet. The existing cells can be moved in two different directions to accomodate the inserted region:

1. Shift Cells Down

2. Shift Cells Right

To insert an entire column or row, simply set the column bottom or row right value to the system maximum:

Rows: 32,000,000
Columns: 32,000

Examples

```
ActiveDocument.NotebookItems(2).InsertCells(0,0,2,99,1)
```

Inserts a block from column 1, row 1 to column 3, row 100 and shifts the current data down.

```
ActiveDocument.NotebookItems("Data 1").InsertCells(0,0,4,32000000,2)
```

Inserts 5 new columns at columns 1- 5 and shifts adjacent columns to the right.

Interpolate3DMesh Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem/ExcelItem object.*Interpolate3Dmesh(*required parameters long,* *optional parameters variants*)

Converts unsorted xyz triplet data to evenly incremented mesh data, as required by mesh and contour plots. The optional parameters control the results columns, mesh range and increment, and original datapoint weighting. Note that the output columns must be specified if the data is to be returned to the worksheet.

Parameters

Required

x input
y input
z input

Optional	**Default Value**
x output	required for results
y output	required for results
z output	required for results
x minimum	Default data min
x maximum	Default data max
y minimum	Default data min
y maximum	Default data max
x intervals	Default 15
y intervals	Default 15
weight	Default 3

Example

```
ActiveDocument.NotebookItems("Data 1").Interpolate3DMesh(0,1,2,3,4,5)
```

Interpolates the data in columns 1, 2 and 3, and places them in columns 4, 5 and 6, using the default values for all other parameters.

IsRegionWriteProtected Method

Objects

Type: Property Get
Result: Boolean

Syntax: *NativeWorksheetItem object*.IsRegionWriteProtected (*left column variant, optional right column variant, optional top row variant, optional bottom row variant*)

Returns whether the specified worksheet region is write protected.

ItalicFont Method

Objects

Type: Sub

Syntax: *ReportItem object*.ItalicFont

Toggles the italic font effect for the selected text.

See the ReportItem object for an example of selection and formatting.

Item Method

Objects

Type: Function
Result: Object

Syntax: *collection*.Item(*object index*)

Returns an object from the collection as specified by the object index number or name. Note that the index begins with 0 by default. The Item method is equivalent to specifying an object from the collection object using an index. If the item does not exist, an error is returned.

Example

```
Dim SelectedPage As Object
Set SelectedPage = Notebooks.Item("Graph Page 1")
```

Sets the notebook item "Graph Page 1" to the object variable SelectedPage. An alternate way of specifying using the Item method is to simply omit the Item function:

```
Set SelectedPage = Notebooks("Graph Page 1")
```

LaunchWizard Method

Objects

Type: Function
Results: Boolean

Syntax: *GraphWizard object*.LaunchWizard

This method launches (opens) the SigmaPlot graph wizard.

Example

```
Dim SPWizard As Object
Set SPWizard = ActiveDocument.CurrentDataItem.GraphWizard

SPWizard.LaunchWizard
```

ModifyWizardPlot Method

Objects

Type: Function

Results: Boolean

Syntax: *GraphItem object*.ModifyWizardPlot(*required parameters variants, optional parameters variants*)

Modifies the current plot on the specified GraphItem object using the following parameters:

Parameter	Values	Optional
graph type	any valid type name	no
graph style	any valid style name	no
data format	any valid data format name	no
column array	any column number/title array	no
columns per plot array	array of columns in each plot	yes
error bar source	any valid source name	error bar plots only
error bar computation	any valid computation name	error bar plots only
anglular axis units	any valid angle unit name	polar plots only
lower range bound	any valid degree value	polar plots only
upper range bound	any valid degree value	polar plots only
ternary units	upper range of ternary axis scale	ternary plots only
lower error bar computation	any valid computation name	error bar plots only
row selection	Boolean: True allows selection of a row range for y-replicate (row-summary) plots. Use False to support pre-y replicate data format macros.	Row summary plots only

Example

```
' Declare an array to hold the columns and start and stop indices.
Dim ColumnsPerPlot()
Redim ColumnsPerPlot(2, 1)
ColumnsPerPlot(0, 0) = 0
ColumnsPerPlot(1, 0) = 0
ColumnsPerPlot(2, 0) = 0
ColumnsPerPlot(0, 1) = 1
ColumnsPerPlot(1, 1) = 0
ColumnsPerPlot(2, 1) = 0
' Declare an array to hold the number of columns per plot.
Dim PlotColumnCountArray()
ReDim PlotColumnCountArray(0)
PlotColumnCountArray(0) = 2 ' We are only adding one plot.
ActiveDocument.CurrentPageItem.ModifyWizardPlot("Vertical Bar Chart", _

"Stacked Bars", _
"Many Y", _
ColumnsPerPlot, _
PlotColumnCountArray, _
"Worksheet Columns", _
"Standard Deviation", _
"Degrees", _
0.000000, _
360.000000)
```

Transforms the current plot into a vertical bar chart.

NormalizeTernaryData Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem/ExcelItem object.* NormalizeTernaryData (*required parameters long, optional parameters variants*)

Normalize three columns of raw data to 100 or 1 for a ternary plot.

Required Parameters	
x input	
y input	
z input	
Optional Parameters	**Default Value**
x output	First Empty
y output	First Empty
z output	First Empty
scale type	100

Example

```
ActiveDocument.NotebookItems("Data 1").NormalizeTernaryData (0,1,2,3,4,5,1)
```

Normalizes the data in columns 1, 2 and 3, and places them in columns 4, 5 and 6, using the normalization to a range of 0-1.

Open Method

Objects

Type: Function
Result: Object

Syntax: *Notebooks collection/NotebookItems object*.Open(*open parameters*)

Opens the notebook specified within the Notebooks collection, or the specified notebook item. The parameter depends upon whether you are opening a notebook or a notebook item.

Notebook	*file name string, optional extension string, optional visible boolean*
NotebookItems None	
PlotEquation	*equation name string*

Note that for NotebookItems and SectionItems, an Open corresponds to an Expanded = True.

Examples

```
Dim NewTemplate As String
NewTemplate = Path " Internat.jnt"

Notebooks.Open(NewTemplate)
```

Opens the Internat.jnt template notebook file.

```
ActiveDocument.Notebooks("Data 2").Open
```

Opens the "Data 2" notebook item.

Paste Method

Objects

Type: Sub

Syntax: *NotebookItems object*.Paste(*format variant*)

Place the contents of the Windows Clipboard into the selected notebook item document, at the current position, if applicable. The format specified is an available clipboard format, as displayed by the Edit menu Paste Special command.

Example

```
Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim index2
index2=0
Dim DataList$(ActiveDocument.NotebookItems.Count)
Dim Item
For Each Item In ActiveDocument.NotebookItems
If ActiveDocument.NotebookItems(Index).ItemType = 1 Or ActiveDocument.NotebookItems(Index).ItemType = 8 Then

DataList$(Index2) = ActiveDocument.NotebookItems(Index).Name
index2=index2+1
End If
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Worksheet Items in Active Notebook" ' %GRID:10,7,1,1
OKButton 210,14,90,21
ListBox 20,14,170,91,DataList(),.ListBox1
End Dialog
Dim dlg1 As UserDialog
Dialog dlg1
Dim SelectedDataSheet
SelectedDataSheet=dlg1.ListBox1
ActiveDocument.NotebookItems(DataList(CLng(SelectedDataSheet))).Open
Dim MaxColumn As Long
Dim MaxRows As Long
MaxColumn = 0
MaxRows = 0
ActiveDocument.CurrentDataItem.DataTable.GetMaxUsedSize(MaxColumn,MaxRows)
Dim collist$()
ReDim collist$(MaxColumn+1)
Dim i
For i=1 To MaxColumn+1
collist$(i)=CStr(i)
Next i
Begin Dialog UserDialog 500,133,"Paste Column" ' %GRID:10,7,1,1
Text 20,21,360,21,"Target Worksheet: "+ActiveDocument.CurrentDataItem.Name,.Text1
OKButton 400,14,80,21
DropListBox 140,77,60,80,collist$(),.DropListBox1
Text 20,77,110,14,"Paste in Column:",.Text2
GroupBox 230,56,220,70,"Paste Behavior",.GroupBox1
OptionGroup .Group1
OptionButton 250,77,20,14,"OptionButton1",.OptionButton1
OptionButton 250,98,20,14,"OptionButton2",.OptionButton2
Text 280,77,160,14,"Shift existing cells down",.Text3
Text 280,98,160,14,"Overwrite existing cells",.Text4
End Dialog
Dim dlg2 As UserDialog
dlg2.DropListBox1=CStr(MaxColumn)
Dialog dlg2
ActiveDocument.CurrentDataItem.Goto(0,dlg2.DropListBox1)
If dlg2.Group1=0 Then
```

```
ActiveDocument.CurrentDataItem.InsertionMode = True
End If
ActiveDocument.CurrentDataItem.Paste
ActiveDocument.NotebookItems("Data 1").InsertionMode= False
```

Pastes the clipboard contents into the specified worksheet column, allowing for inserting or overstriking the current column contents.

PlotEquation Method

Type: Sub
Result: Object

Syntax: *GraphItem object*.PlotEquation

Returns a PlotEquation object for graphing equation data.

Example

```
Sub Main
Dim SPEquation As Object
Set SPEquation = ActiveDocument.CurrentPageItem.PlotEquation

SPEquation.EquationRHS = "95*exp(-.5*((x)/2)^2)"
SPEquation.Plot
End Sub
```

Plots the equation $y = 95e-5(x/2)^2$.

Print Method

Objects

Type: Sub

Syntax: *Notebook/NotebookItems object*.Print(*printer port string*)

Prints the selected item, including any items within specified NotebookItems and SectionItems. Specifying the Notebook prints all items in the notebook.

Example

```
Dim DefaultPrinter As String
DefaultPrinter = \\FILESERVER1\LaserPrinter
ActiveDocument.NotebookItems("Graph Page 1").Print(DefaultPrinter)
```

Prints the page "Graph Page 1" to the printer with the printer port of \\FILESERVER1\LaserPrinter.

PrintStatsWorksheet Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem object*.PrintStatsWorksheet

Prints the NativeWorksheetItem’s statistics worksheet. If the worksheet has not been opened using the ShowStatsWorksheet property, this method fails.

Example

```
Activedocument.CurrentDataItem.ShowStatsWorksheet=True

Activedocument.CurrentDataItem.PrintStatsWorksheet
```

Prints column statistics for the current worksheet.

PutData Method

Objects

Type: Sub

Syntax: *DataTable object*.PutData(*array* variant,*left* long, *top* long)

Places the specified array variant into the worksheet starting at the specified location. The data can be a 2D array.

Example

```
Dim Data(1,4) As Variant
Data(0,0) = "A"
Data(0,1) = "B"
Data(0,2) = "C"
Data(0,3) = "D"
Data(0,4) = "E"
Data(1,0) = 1
Data(1,1) = 7
Data(1,2) = 3
Data(1,3) = 4
Data(1,4) = 9
ActiveDocument.CurrentDataItem.DataTable.PutData(Data,0,0)
```

Places the 2D array variable "Data" into the "Data 1" worksheet, beginning at cell 1, 1.

Quit Method

Objects

Type: Sub

Syntax: Quit

Ends SigmaPlot. If SigmaPlot is in use, then this method is ignored.

Redo Method

Objects

Type: Sub

Syntax: *object*.Redo

Redoes the last undone action for the specified object. If redo has been disabled in SigmaPlot for either the worksheet or page, this method has no effect.

Example

```
ActiveDocument.NotebookItems("Graph Page 1").Redo
```

This undoes the last user "Undo" on "Graph Page 1".

Remove Method

Objects

Type: Function
Result: Boolean

Syntax: *NamedDataRanges/GraphObject collection*.Remove(*index variant*)

Deletes the specified object. The index can be a number or a name. If the specified index does not exist, an error is returned.

Examples

```
ActiveDocument.CurrentDataItem.DataTable.NamedRanges.Remove("Title 1")
```

Removes the NamedDataRange "Title 1" from the data table of the current worksheet.

```
ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Remove(0)
```

Removes the first item on the current page.

Run Method

Objects

Type: Function
Result: Boolean

Syntax: *MacroItem/FitItem object*.Run

Runs a FitItem or Macro without closing the object.

Example

```
Dim Selection(3)
Selection(0) = 0
Selection(1) = 0
Selection(2) = 1
Selection(3) = &H7FFFFFF
```

```
ActiveDocument.CurrentDataItem.SelectionExtent = Selection
Dim ActiveDoc As Object
Dim CurItem As Object
Set ActiveDoc = ActiveDocument
Set CurItem = ActiveDocument.CurrentItem
Notebooks.Open(path+"\Standard.jfl")
ActiveDoc.Activate
CurItem.IsCurrentItem = True
Dim FitObject As Object
Set FitObject = Notebooks(path+"\Standard.jfl").NotebookItems("Quadratic")

FitObject.Open
FitObject.Variable("x") = "col(1)"
FitObject.Variable("y") = "col(2)"
FitObject.Run
FitObject.Finish
```

Fits a quadratic curve to the data in the first two columns of the current worksheet.

The following example shows a complete run of the fit wizard as the macro recorder records it (excluding the comments). It contains "Run" and "Finish" as well as the various statements needed to set up a curve fit session.

```
' Remember the current item and document
Dim CurItem As Object
Set CurItem = ActiveDocument.CurrentItem
Dim ActiveDoc As Object
Set ActiveDoc = ActiveDocument

' Open the fit file containing the fit we want to run.
Notebooks.Open(path+"\Standard.jfl")
Dim FitFile As Object
Set FitFile = Notebooks("C:\Data\PROJ\spw32\Standard.jfl")

' Reset the current document and worksheet to get fit data from.
ActiveDoc.Activate
CurItem.IsCurrentItem = True

' Open the fit we want to run.
Dim FitObject As Object
Set FitObject = Notebooks(path+"\Standard.jfl").NotebookItems("Single, 2 Parameter")

FitObject.Open

' Set the data format and set the variables
FitObject.DatasetType = CF_XYPAIR
FitObject.Variable("x") = "col(1)"
FitObject.Variable("y") = "col(2)"

' Run the fit. (This computes the fit results but
' does not output graphs, data, or reports.
FitObject.Run
' Set the output parameters
FitObject.OutputReport = False
FitObject.OutputEquation = False
FitObject.ResidualsColumn = -2
FitObject.PredictedColumn = -2
FitObject.ParametersColumn = -2
FitObject.OutputGraph = True
FitObject.OutputAddPlot = True
FitObject.AddPlotGraphIndex = -1
FitObject.XColumn = -1
FitObject.YColumn = -1
FitObject.ZColumn = -2

' Output the results (this would also "Run" the fit if
' we hadn't already done that.
FitObject.Finish

' Close the fit file and set the variable to "Nothing" to make sure the
' fit file is completely released. (We would not be able to reopen it
' until this is done or this macro finishes).
```

```
FitFile.Close(True)
Set FitFile = Nothing
```

RunEditor Method

Objects

Type: Sub

Syntax: *TransformItem object*.RunEditor

Invokes the user defined transform editor for the specified transform item.

Example

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)

SPTransform.Name = path + "\Transforms\Anova.xfm"
SPTransform.Open
SPTransform.RunEditor
```

Opens the ANOVA transform for editing.

SaveAs Method

Objects

Type: Sub

Syntax: *Notebook object*.SaveAs(*file name string*)

Save a notebook file for the first time, or to a new file name and path. Note that you need to provide the file extension. Recognized SigmaPlot notebook file extensions are .JNB, .JNT, and .JFL

Example

```
Dim FileName As String
FileName = "d:\My Documents\My Notebook.jnb"

ActiveDocument.SaveAs(FileName)
```

Saves the currently active notebook to the file name and path d:\My Documents\My Notebook.jnb

Save Method

Objects

Type: Sub

Syntax: *Notebook/NotebookItems object*.Save

Saves a Notebook object to disk using the current FullName , or a notebook item to the notebook (without saving the notebook file to disk). If no FullName exists for a notebook, an error occurs. To save a notebook that has not yet been saved, you must use the SaveAs method.

Note: Transform text can be saved to an .xfm file by naming the transform first with the full file name, extension, and path.

Examples

```
Notebooks("c:\My Documents\My Notebook.jnb").Save
```

Saves the notebook "My Notebook.jnb."

```
ActiveDocument.NotebookItems("Graph Page 1").Save
```

Updates the version of the "Graph Page 1" page in the notebook.

Select Method

Objects

Type: Sub

Syntax: *GraphItem object*.Select *AddToSelection:=variable boolean, Left:= variable variant, _*
Top:= variable variant, Right:= variable variant, Bottom:= variable variant

Selects all of the items within the specified selection region. In addition, if "*Top*" equals "*Bottom*" and "*Right*" equals "*Left*", the resulting selection includes the object that the specified point lies within.

If "*AddToSelection*" is "False" then the previous selection list is replaced by the new list. If "True", then the newly selected items are added to the existing selection list.

Examples

```
ActiveDocument.CurrentPageItem.Select(False, -5500, 4062, -5500, 4062)
ActiveDocument.CurrentPageItem.Select(True, -1375, 875, -1375, 875)
ActiveDocument.CurrentPageItem.Select(True, 2062, 1208, 2062, 1208)
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SEA_THICKNESS, 39)

ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SEA_LINETYPE, 5)
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SEA_END2TYPE, 4)
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SEA_END1TYPE, 3)
```

Selects an item on the graph page (a line in this case) and adjusts the thickness, type and endpoint appearance. (The location of your objects will vary from the specified coordinates.)

```
Dim GraphPage As Object
Set GraphPage =ActiveDocument.NotebookItems("Graph Page 1")

GraphPage.Select(False,3500,1750,5000,3500)
GraphPage.Export("c:\Mygraph.bmp","BMP")
```

Selects the graph displayed on Graph Page 1 and exports the image to a bitmap file.

SelectAll Method

Objects

Type: Sub

Syntax: *object*.SelectAll

Selects the entire contents of the item.

Examples

```
ActiveDocument.CurrentDataItem.SelectAll
Dim cname As String
cname=ActiveDocument.CurrentDataItem.Name
ActiveDocument.CurrentDataItem.Copy
ActiveDocument.NotebookItems.Add(1)
ActiveDocument.CurrentDataItem.Name="Copy of " + cname

ActiveDocument.CurrentDataItem.Paste
```

Creates a copy of the current worksheet.

```
Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim index2
index2=0
Dim ReportList$(ActiveDocument.NotebookItems.Count)
Dim Item
For Each Item In ActiveDocument.NotebookItems
If ActiveDocument.NotebookItems(Index).ItemType = 5 Then
ReportList$(Index2) = ActiveDocument.NotebookItems(Index).Name
index2=index2+1
End If
Index = Index + 1
Next Item
Begin Dialog UserDialog 320,119,"Report Items in Active Notebook" ' %GRID:10,7,1,1

OKButton 210,14,90,21
ListBox 20,14,170,91,ReportList(),.ListBox1
End Dialog
Dim dlg1 As UserDialog
Dialog dlg1
Dim selreport
selreport=dlg1.ListBox1
ActiveDocument.NotebookItems(ReportList$(dlg1.ListBox1)).Open
ActiveDocument.CurrentItem.SelectAll
ActiveDocument.CurrentItem.Copy
ActiveDocument.NotebookItems.Add(1)
ActiveDocument.CurrentItem.Paste
```

Pastes the entire contents of the selected report into a new worksheet. Hard returns in the copied text define new rows in the worksheet. Tabs define new columns.

SelectObject Method

Objects

Type: Sub

Syntax: *object*.SelectObject

Clears the current GraphItem selection list and selects the specified graph object so that it can be altered using the SetSelectedObjectsAttribute method. Line and Solid objects can only be selected if they are top level drawing objects (not child objects of other objects).

Example

The following example selects each of two graphs to allow using SetSelectedObjectsAttribute to change their colors.

```
If ActiveDocument.CurrentPageItem.GraphPages(0).Graphs.Count > 1 Then
' Select the first graph on the page and turn it red. ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).SelectObject

ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SOA_COLOR,RGB_RED)
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(STA_COLOR,RGB_RED)
' Select the second graph on the page and turn it green.
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(1).SelectObject
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(SOA_COLOR,RGB_GREEN)
ActiveDocument.CurrentPageItem.SetSelectedObjectsAttribute(STA_COLOR,RGB_GREEN)
Else
MsgBox("This macro requires 2 graphs on the page.",0+48,"Error")
End If
```

SetAttribute Method

Objects

Type: Function
Result: Long

Syntax: *Page object/child object.*SetAttribute(*attribute, parameter*)

The SetAttribute method is used by all graph page objects to change current attribute settings. Attributes are numeric values that also have constants assigned to them. For a list of all these attributes and constants, see **SigmaPlot Constants**.

Message Forwarding: If you use the SetAttribute method to change an attribute that does not exisit for the current object, the message is automatically routed to an object that has this attribute using the message forwarding table.

Using the Object Browser to view Constants You can view alternate values for attributes and constants by selecting the current attribute value, then clicking the Object Browser button. All valid alternate values will be listed—to use a different value, select the value and click Paste.

Examples

```
ActiveDocument.CurrentPageItem.GraphPages(0).Graphs(0).Plots(0).SetAttribute(SLA_TYPE,SLA_TYPE_BAR)
```

Converts the first plot in the first graph on the first graph page to a bar chart.

```
Dim Points()
Redim Points(3)
Points(0) = -2854
Points(1) = -354
Points(2) = -542
Points(3) = -2145
Dim LineObject As Object
Set LineObject = ActiveDocument.CurrentPageItem.GraphPages(0).ChildObjects.Add(GPT_LINE, Points)

LineObject.SetAttribute(SEA_END2TYPE, 2)
```

Draws an arrow on the current graph page.

SetCurrentObjectAttribute Method

Objects

Type: Function
Result: Long

Syntax: *GraphItem object*.SetCurrentObjectAttribute(*attribute variant*,*property variant*,*setting variant*)

Changes the attribute specified by *attribute* of the current object on the graphics page. Use the Set Attribute Constants to specify the attribute argument. This method most often appears in recorded macros.

The properties available for the current object are entirely dependent on the type of object.

Use one of the following three techniques to set the current object on the graphics page:

- Click the object using the mouse
- Use the SigmaPlot menus (e.g. "Select Graph")
- Use the SetObjectCurrent method

If the specified GraphItem is not open or there is no current object of the appropriate type on the page, the method will fail.

Examples

```
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SSA_SHAPE, 2)
```

Sets symbols in the current plot to circles.

```
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SSA_SIZE, 191)
```

Sets all symbols in the current plot to a size of 191.

SetObjectCurrent Method

Objects

Type: Sub

Syntax: *object*.SetObjectCurrent

Sets the specified object to the "current" object for the purpose of the "SetCurrentObjectAttribute" command.

If the specified GraphItem is not open, this method will fail.

Examples

```
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SLA_SELECTDIM, 1)
ActiveDocument.CurrentPageItem.GraphPages(0).CurrentPageObject(GPT_AXIS).NameObject.SetObjectCurrent
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETAXISATTR, SAA_RTFNAME, "{\rtf1\ansi0{\colortbl\red0
\green0\blue0;}\deff0{\fonttbl\f0\fnil Arial;\f1\fnil Symbol;}{\sl240\slmult0\f0\cf0\up0\fs24\i0\b0\ul0\ql Bottom Axis - \f1d}}")
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETPLOTATTR, SLA_SELECTDIM, 2)
ActiveDocument.CurrentPageItem.GraphPages(0).CurrentPageObject(GPT_AXIS).NameObject.SetObjectCurrent
ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM_SETAXISATTR, SAA_RTFNAME, "{\rtf1\ansi0{\colortbl\red255
\green0\blue0;}\deff0{\fonttbl\f0\fnil Arial;\f1\fnil Symbol;}{\sl240\slmult0\f0\cf0\up0\fs24\i0\b0\ul0\ql Side Axis - \f1s}}")
```

Sets the X-axis title to "Bottom Axis – d" and the Y-axis title to "Side Axis – σ". In addition, the Y-axis title appears as red text.

SetRegionBorderThickness Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem object*. SetRegionBorderThickness(*border thickness long, left column variant, optional right column variant*)

Set the border thickness of the specified worksheet region. These borders are defined as the left-hand border of the region for columns borders, and the row grid lines within that region.

The border thickness argument is an integer that corresponds to which region you want to set to thick borders:

Value	Effect
0	No thick borders
1.	Left side only
2.	Rows only
3.	Rows and left side

Example

The following program can be used to set column and column grid borders.

```
Dim Worksheet As Object
Dim Column As Long
Option Explicit
```

```
Sub Main
Set Worksheet = ActiveDocument.CurrentDataItem
Column = 1
MacroDialog:
Begin Dialog UserDialog 240,154,"Set Border Thickness",.DialogFunc ' %GRID:10,7,1,1

PushButton 10,7,80,21,"Left Side",.LeftSide
PushButton 10,35,80,21,"Right Side",.RightSide
PushButton 10,63,80,21,"Row Grid",.RowsOnly
PushButton 10,91,80,21,"All Borders",.AllBorders
PushButton 10,119,80,21,"Clear All",.ClearAll
Text 120,14,120,14,"Column Number",.Text1
TextBox 120,35,90,21,.ColumnNumber
OKButton 140,119,80,21,.OKButton
End Dialog
Dim dlg As UserDialog
dlg.ColumnNumber = CStr(Column + 1)
Select Case Dialog(dlg)
Case 1
```

```
Column = CLng(dlg.ColumnNumber) - 1
Worksheet.SetRegionBorderThickness(1,Column)
Worksheet.SetRegionBorderThickness(0,Column + 1)
GoTo MacroDialog
Case 2
Column = CLng(dlg.ColumnNumber) - 1
Worksheet.SetRegionBorderThickness(1,Column + 1)
Worksheet.SetRegionBorderThickness(0,Column)
GoTo MacroDialog
Case 3
Column = CLng(dlg.ColumnNumber) - 1
Worksheet.SetRegionBorderThickness(2,Column)
Worksheet.SetRegionBorderThickness(0,Column + 1)
GoTo MacroDialog
Case 4
Column = CLng(dlg.ColumnNumber) - 1
Worksheet.SetRegionBorderThickness(3,Column)
Worksheet.SetRegionBorderThickness(1,Column + 1)
GoTo MacroDialog
Case 5
Column = CLng(dlg.ColumnNumber) - 1
Worksheet.SetRegionBorderThickness(0,Column)
Worksheet.SetRegionBorderThickness(0,Column + 1)
GoTo MacroDialog
End Select
End Sub
Function DialogFunc%(DlgItem$, Action%, SuppValue%)
Select Case Action%
Case 1 ' Dialog box initialization
DlgText "OKButton","Close"
End Select
End Function
```

SetSelectedObjectsAttribute Method

Objects

Type: Function
Result: Long

Syntax: *GraphItem object*.SetSelectedObjectsAttribute(*Attribute*, *Parameter*)

Changes the attribute specified by "*Attribute*" for all the selected objects on the graphics page. Select graphics page objects using one of the following two techniques:

- Click the object with the mouse.
- Use the SelectObject method.

Valid *Attribute* values include:

SOA_COLOR
SEA_LINETYPE
SDA_EDGECOLOR

SEA_THICKNESS
SEA_ENDSIZE
SEA_END1SIZE
SEA_END2SIZE
SEA_LINEEND1
SEA_LINEEND2

SDA_PATTERN
SDA_COLOR
SOA_SIZEEX

SOA_POSEX
STA_FONT
STA_ITALIC
STA_BOLD
STA_UNDERLINE
STA_SIZE

SEA_END1TYPE
SEA_END2TYPE
SEA_END1ANGLE
SEA_END2ANGLE
SEA_END1POINT
SEA_END2POINT

STA_COLOR
STA_LINESPACING
STA_PARAGRAPHJUSTIFY
STA_RELORIENTATION
STA_ORIENTATION

The size and position attributes are likely to work best when only one object is selected.

It the specified GraphItem is not open or if there are no selected objects on the page, the method will fail.

Example

```
Dim ActivePage As Object
Dim ActiveDoc As Object

Set ActiveDoc = ActiveDocument
Set ActivePage = ActiveDoc.CurrentPageItem

ActivePage.SetSelectedObjectsAttribute(SOA_COLOR,RGB_RED)
ActivePage.SetSelectedObjectsAttribute(SEA_LINETYPE,SEA_LINE_DOTTED)
```

Changes the appearance of the selected line to a red, dotted line.

SetTitles Method

Objects

Type: Function
Result:

Syntax: *GraphWizard object*.SetTitles(title list variant)

Sets the list of data variables/columns listed in the graph wizard data picking panel for the GraphWizard object.

Example

```
Dim DataList()
ReDim DataList(4)
DataList(0) = "Group A"
DataList(1) = "Group B"
DataList(2) = "Group C"
DataList(3) = "Group D"
DataList(4) = "Group E"

Dim SPWizard As Object
Set SPWizard = ActiveDocument.CurrentDataItem.GraphWizard

SPWizard.SetTitles(DataList)
SPWizard.LaunchWizard
```

SortSelection Method

Objects

Type: Sub

Syntax: *ExcelItem/NativeWorksheet object*.SortSelection(*key column long*, *start column long*, *start row long*, *end column long*, *start row long*, *direction long*)

Performs a key-column alpha-numeric sort on the specified data region. Note that if you also want to sort the specified key column, you need to include it in the sorted region.

Use a direction valuem of 0 for ascending, or 1 for descending.

Example

```
Dim Key, First, Top, Last, Bottom, Direction As Long
Key = 0
First = 0
Top = 0
Last = 3
Bottom = 31999999
Direction = 0
Dim CurrentWorksheet
Set CurrentWorksheet = ActiveDocument.CurrentDataItem
CurrentWorksheet.SortSelection(Key, First, Top, Last, Bottom, Direction)
```

Sorts the region starting in column 1 through column 4, using column 1 as the key column, in ascending order.

StockScheme Method

Objects

Type: Property Get
Result: Long

Syntax: *Graph object*.StockScheme(*stockscheme long*)

Returns the property scheme value for a variable, which can then be assigned to a graph object.

STOCKSCHEME_COLOR_BW	&H00010001
STOCKSCHEME_COLOR_GRAYS	&H00020001
STOCKSCHEME_COLOR_EARTH	&H00030001
STOCKSCHEME_COLOR_FOREST	&H00040001
STOCKSCHEME_COLOR_OCEAN	&H00050001
STOCKSCHEME_COLOR_RAINBOW	&H00060001
STOCKSCHEME_COLOR_OLDINCREMENT	&H00070001
STOCKSCHEME_SYMBOL_DOUBLE	&H00010002
STOCKSCHEME_SYMBOL_MONOCHROME	&H00020002
STOCKSCHEME_SYMBOL_DOTTEDDOUBLE	&H00030002

STOCKSCHEME_SYMBOL_OLDINCREMENT	&H00040002
STOCKSCHEME_LINE_MONOCHROME	&H00010003
STOCKSCHEME_LINE_OLDINCREMENT	&H00020003
STOCKSCHEME_PATTERN_MONOCHROME	&H00010004
STOCKSCHEME_PATTERN_OLDINCREMENT	&H00020004

Example

```
Dim ANotebook As Object
Set ANotebook = Notebooks.Add
Dim DataItem As Object
Set DataItem = ANotebook.NotebookItems("Data 1")
Dim ADataTable As Object
Set ADataTable = DataItem.DataTable

' Create some example data.
Dim i
For i = 1 To 5
ADataTable.Cell(0,i-1) = i
ADataTable.Cell(1,i-1) = i+1
ADataTable.Cell(2,i-1) = i+2
ADataTable.Cell(3,i-1) = i+3
ADataTable.Cell(4,i-1) = i+4
Next i
Dim Sign
Sign = 1
For i = 1 To 5
ADataTable.Cell(5,i - 1) = 100 + i*Sign
Sign = -Sign
Next i

Dim GraphicPage
Set GraphicPage = ANotebook.NotebookItems.Add(CT_GRAPHICPAGE)
'Create a graph manually. (This isn't recommended. Better to use CreateWizardGraph)
Dim PageObject As Object
Set PageObject = GraphicPage.GraphPages(0)
Dim AGraphObject As Object
Set AGraphObject = PageObject.ChildObjects.Add(GPT_GRAPH, SGA_COORD_CART2, SLA_TYPE_BAR, SLA_SUBTYPE_VERTY)

Dim PlotObject As Object
Set PlotObject = AGraphObject.Plots(0)
' Plot objects only allow you to add objects of type GPT_TUPLE
' Add 4 tuples to make a grouped bar chart with groups of 4.
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,1)
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,2)
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,3)
PlotObject.ChildObjects.Add(GPT_TUPLE, 0,4)
' Get some repeat type schemes for the two tuples.
Dim FillScheme
FillScheme = PlotObject.StockScheme(STOCKSCHEME_PATTERN_OLDINCREMENT)
' Tell the plot to use the "old increment" scheme"
PlotObject.Fill.SetAttribute(SDA_PATTERNREPEAT, FillScheme)
' Set the initial density and pattern
PlotObject.Fill.SetAttribute(SDA_PATTERN, (SDA_DENS_FINE*&H10000) + SDA_PAT_HOLLOW)
' Get some repeat type schemes for the two tuples.
Dim ColorScheme
ColorScheme = PlotObject.StockScheme(STOCKSCHEME_COLOR_GRAYS)
' Tell the plot to use the "gray" scheme"
PlotObject.Fill.SetAttribute(SDA_COLORREPEAT, ColorScheme)
' Set the initial color in the pattern
PlotObject.Fill.SetAttribute(SDA_COLOR, RGB_GRAY)
```

Creates a bar graph for some generated data and applies the Gray stockscheme to the result.

TransposePaste Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem object*.TransposePaste

Pastes the data in the clipboard into the worksheet, transposing the row and column indices of the data such that rows and columns are swapped. If there is nothing in the clipboard or the data is not of the right type, nothing will happen.

Examples

```
ActiveDocument.CurrentDataItem.TransposePaste
```

Pastes the clipboard contents into the current worksheet such that the rows become columns and the columns become rows.

```
Begin Dialog UserDialog 400,126,"Copy and Paste" ' %GRID:10,7,1,1
OKButton 310,14,70,21
GroupBox 20,42,270,70,"Copy",.GroupBox1
Text 20,14,270,21,"Current Worksheet: "+ activedocument.CurrentDataItem.Name,.Text1
OptionGroup .Group1
OptionButton 40,56,80,14,"Column",.OptionButton1
OptionButton 40,84,90,14,"Row",.OptionButton2
TextBox 240,70,40,21,.TextBox1
Text 140,70,100,14,"Index number:",.Text2
End Dialog
Dim dlg1 As UserDialog
dlg1.TextBox1="1"
Dialog dlg1
Dim Selection(3)
If dlg1.Group1=0 Then
Selection(0) = CLng(dlg1.TextBox1)-1
Selection(1) = 0
Selection(2) = CLng(dlg1.TextBox1)-1
Selection(3) = &H7FFFFFF
ActiveDocument.CurrentDataItem.SelectionExtent = Selection
ActiveDocument.CurrentDataItem.Copy
Else
Selection(0) = 0
Selection(1) = CLng(dlg1.TextBox1)-1
Selection(2) = &H7FFFFFF
Selection(3) = CLng(dlg1.TextBox1)-1
ActiveDocument.CurrentDataItem.SelectionExtent = Selection
ActiveDocument.CurrentDataItem.Copy
End If

Dim NotebookItems$()
ReDim NotebookItems$(ActiveDocument.NotebookItems.Count)
Dim Index
Index = 0
Dim index2
index2=0
Dim DataList$(ActiveDocument.NotebookItems.Count)
Dim Item
For Each Item In ActiveDocument.NotebookItems
If ActiveDocument.NotebookItems(Index).ItemType = 1 Or ActiveDocument.NotebookItems(Index).ItemType = 8 Then

DataList$(Index2) = ActiveDocument.NotebookItems(Index).Name
index2=index2+1
End If
Index = Index + 1
```

```
Next Item

Begin Dialog UserDialog 420,238,"Paste" ' %GRID:10,7,1,1
Text 20,14,120,14,"Target Worksheet:",.Text1
DropListBox 160,14,150,84,DataList(),.DropListBox1
OKButton 330,14,70,21
GroupBox 20,49,280,77,"Paste Location",.GroupBox1
OptionGroup .Group1
OptionButton 40,70,80,14,"Column",.OptionButton1
OptionButton 40,98,80,14,"Row",.OptionButton2
Text 140,84,100,21,"Index number:",.Text2
TextBox 250,84,40,21,.TextBox1
GroupBox 20,140,280,77,"Paste Behavior",.GroupBox2
OptionGroup .Group2
OptionButton 40,161,220,14,"Shift existing cells down",.OptionButton3
OptionButton 40,182,170,21,"Overwrite existing cells",.OptionButton4
End Dialog
Dim dlg2 As UserDialog
dlg2.TextBox1=dlg1.TextBox1
Dialog dlg2
If dlg2.Group2=0 Then
ActiveDocument.CurrentDataItem.InsertionMode = True
End If
ActiveDocument.NotebookItems(DataList(CLng(dlg2.DropListBox1))).Open
If dlg2.Group1=0 Then
ActiveDocument.CurrentDataItem.Goto(0,CLng(dlg2.TextBox1)-1)
If dlg1.Group1=0 Then
ActiveDocument.CurrentDataItem.Paste
Else
ActiveDocument.CurrentDataItem.TransposePaste
End If
Else
ActiveDocument.CurrentDataItem.Goto(CLng(dlg2.TextBox1)-1,0)
If dlg1.Group1=0 Then
ActiveDocument.CurrentDataItem.TransposePaste
Else
ActiveDocument.CurrentDataItem.Paste
End If
End If
ActiveDocument.NotebookItems("Data 1").InsertionMode= False
```

Copies a row or column from a worksheet and pastes the copied entries as a row or column in the specified worksheet.

UnderlineFont Method

Objects

Type: Sub

Syntax: *ReportItem object*.UnderlineFont

Toggles the underline font effect for the selected text.

See the ReportItem object for an example of selection and formatting.

Undo Method

Objects

Type: Sub

Syntax: *object*.Undo

Undoes the last performed action for the specified object. If undo has been disabled in SigmaPlot for either the worksheet or page, this method has no effect.

Example

```
ActiveDocument.NotebookItems("Graph Page 1").Undo
```

Undoes the last user action on "Graph Page 1".

WriteProtectRegion Method

Objects

Type: Sub

Syntax: *NativeWorksheetItem object*.WriteProtectRegion(*toggle boolean, left column variant, optional right column variant, optional top row variant, optional bottom row variant*)

Write-protect the specified worksheet region.

Example

```
Dim Worksheet As Object
Dim FirstColumn, LastColumn As Long
Option Explicit
```

```
Sub Main
Set Worksheet = ActiveDocument.CurrentDataItem
FirstColumn = 0
LastColumn = 1
MacroDialog:
Begin Dialog UserDialog 280,98,"Write Protect Columns", .DialogFunc ' %GRID:10,7,1,1

PushButton 10,70,80,21,"Protected",.PushButton1
PushButton 100,70,80,21,"Editable",.PushButton2
PushButton 190,70,80,21,"Test",.PushButton3
OKButton 190,7,80,21,.OKButton
Text 10,10,90,21,"Start Column",.Text1
TextBox 110,7,70,21,.StartCol
Text 10,38,80,21,"End Column",.Text2
TextBox 110,35,70,21,.EndCol
End Dialog
Dim dlg As UserDialog
dlg.StartCol = CStr(FirstColumn + 1)
dlg.EndCol = CStr(LastColumn + 1)
Select Case Dialog(dlg)
Case 1
FirstColumn = CLng(dlg.StartCol) - 1
LastColumn = CLng(dlg.EndCol) - 1
Worksheet.WriteProtectRegion(True,FirstColumn,LastColumn)
GoTo MacroDialog
Case 2
FirstColumn = CLng(dlg.StartCol) - 1
LastColumn = CLng(dlg.EndCol) - 1
Worksheet.WriteProtectRegion(False,FirstColumn,LastColumn)
GoTo MacroDialog
Case 3
Dim Title$
Title = "Are columns "+dlg.StartCol+" through "+dlg.EndCol+" write protected?"
MsgBox Worksheet.IsRegionWriteProtected(FirstColumn,LastColumn), Title
GoTo MacroDialog
End Select
End Sub
Function DialogFunc%(DlgItem$, Action%, SuppValue%)
Select Case Action%
Case 1 ' Dialog box initialization
DlgText "OKButton","Close"
End Select
End Function
```

FitItem and FitResults Properties and Methods

For examples, see the FitItem and FitResults objects.

FitItem Properties

AddPlotGraphIndex

Constraint

DatasetType

DependentVariableName

Equation

FitResults

FittedParameterValue

Option

OutputAddPlot

OutputEquation

OutputGraph

OutputReport

Parameter

ParametersColumn

PredictedColumn

ResidualsColumn

TrigUnit

Variable

WeightVariableName

XColumn

YColumn

ZColumn

FitItem Methods

Finish

IterateMore

Run

FitResults Properties

AdjustedRSquare

DataPointCount

DurbinWatson

FValue

HasWeights

KolmogorovSmirnovPValue

MissingCount

ParameterCount

PerfectFit

PRESS

PValue

RegressionDegreesOfFreedom

RegressionSumOfSquares

ResidualDegreesOfFreedom

ResidualSumOfSquares

RSquare

RValue

SpearmanRValue

StandardErrorOfEstimate

TotalDegreesOfFreedom

TotalSumOfSquares

FitResults Methods

ConfidenceLimitPopulationLower

ConfidenceLimitPopulationUpper

ConfidenceLimitRegressionLower

ConfidenceLimitRegressionUpper

FitVerdict

OriginalObservationIndex

ParameterDependency

ParameterPValue

ParameterRegressionCoefficient

ParameterStandardError

ParameterTValue

Power

PredictedValue

ResidualValue

StandardizedResidual

StudentizedDeletedResidual

StudentizedResidual

AddPlotGraphIndex Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.AddPlotGraphIndex

Sets/returns the index for the graph used for plotting the curve fit results.

For examples, see the FitItem and FitResults objects.

AdjustedRSquare Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.AdjustedRSquare

Returns the Adjusted *R* Square computed for the regression.

For examples, see the FitItem and FitResults objects.

ConfidenceLimitPopulationLower Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ConfidenceLimitPopulationLower(*observation index long*)

Returns the lower confidence limit for the population, for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

ConfidenceLimitPopulationUpper Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ConfidenceLimitPopulationUpper(*observation index long*)

Returns the upper confidence limit for the population, for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

ConfidenceLimitRegressionUpper Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ConfidenceLimitRegressionUpper(*observation index long*)

Returns the upper confidence limit for the regression, for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

ConfidenceLimitRegressionLower Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ConfidenceLimitRegressionLower(*observation index long*)

Returns the lower confidence limit for the regression, for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

Constraint Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.Constraint(*index long*)

Returns the constraint value specified by the index.

For examples, see the FitItem and FitResults objects.

DataPointCount Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.DataPointCount

Returns the number of original datapoints.

For examples, see the FitItem and FitResults objects.

DatasetType Property

Objects

Read/Write
Value: Long

Syntax: *FitItem object*.DatasetType

Returns the data format used by the curve fit, using one of the following values:

CF_XYPAIR	0
CF_XYZTRIPLET	1.
CF_FROMCODE	2.
CF_SINGLEY	3.
CF_XMANYY	4.
CF_MANYY	5.
CF_XYMANYZ	6.
CF_MANYZ	7.
CF_XMANYINDEPENDENT	8.

For examples, see the FitItem and FitResults objects.

DependentVariableName Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.DependentVariableName

Sets or returns the name of the dependent variable (typically "y").

For examples, see the FitItem and FitResults objects.

DurbinWatson Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.DurbinWatson

Returns the Durbin-Watson statistic computed for the regression.

For examples, see the FitItem and FitResults objects.

Equation Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.Equation(*name*)

Specifies the current equation and equation name.

For examples, see the FitItem and FitResults objects.

ExtendFitToAxes Property

Objects

Read/Write
Value: Boolean

Syntax: *FitItem object*.ExtendFitToAxes

Returns/sets whether or not the curve fit graph output extends to the axes.

For examples, see the FitItem and FitResults objects.

Finish Method

Objects

Type: Sub

Syntax: *FitItem object*.Finish

Close the wizard and execute the fit.

For examples, see the FitItem and FitResults objects.

FitResults Property

Objects

Read/Write
Value: Object

Syntax: *FitItem object*.FitResults

Returns the FitResults object for the FitItem.

For examples, see the FitItem and FitResults objects.

FittedParameterValue Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.FittedParameterValue(*name*)

Returns the value of the last fitted parameters for the current FitItem, for the specified parameter name.

For examples, see the FitItem and FitResults objects.

FitVerdict Method

Objects

Type: Property Get
Result Double

Syntax: *FitResults object*.FitVerdict(*parameter name variant*)

Returns the curve fit verdict value. See Curve Fitter Verdicts Constants for possible return values.

For examples, see the FitItem and FitResults objects.

FValue Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.FValue

Returns the *F* value computed for the regression.

For examples, see the FitItem and FitResults objects.

HasWeights Property

Objects

Read Only
Value: Boolean

Syntax: *FitResults object*.HasWeights

Returns whether weighting was used or not for the regression.

For examples, see the FitItem and FitResults objects.

IterateMore Method

Objects

Type: Sub

Syntax: *FitItem object*.IterateMore

Continue with more iterations if the number of iterations specified by the Option property is exceeded.

For examples, see the FitItem and FitResults objects.

KolmogorovSmirnovPValue Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*. KolmogorovSmirnovPValue

Returns the *P* value for the KolmogorovSmirnov (normality) test computed for the regression.

For examples, see the FitItem and FitResults objects.

MissingCount Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.MissingCount

Returns the number of missing values in the dataset.

For examples, see the FitItem and FitResults objects.

Option Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.Option(*name variant*)

Sets/returns the value of the the specified option. The curve fit options are

Option Name	Default Value
Iterations	100
Stepsize	100
Tolerance	0.0001

For examples, see the FitItem and FitResults objects.

OriginalObservationIndex Method

Objects

Type: Property Get

Result: Long

Syntax: *FitResults object*.OriginalObservationIndex(*observation number long*)

Returns the index for the given observation number.

For examples, see the FitItem and FitResults objects.

OutputAddPlot Property

Objects

Read/Write
Value: Boolean

Syntax: *FitItem object*.OutputAddPlot

Determines whether or not the curve fit results are plotted by adding a plot to the specified output graph. The specified graph is determined by the AddPlotGraphIndex property.

For examples, see the FitItem and FitResults objects.

OutputEquation Property

Objects

Read/Write
Value: Boolean

Syntax: *FitItem object*.OutputEquation

Determines whether or not a copy of the current equation is added to the data section of the target notebook.

For examples, see the FitItem and FitResults objects.

OutputGraph Property

Objects

Read/Write
Value: Boolean

Syntax: *FitItem object*.OutputEquation

Determines whether or not the fit results are plotted on a new graph.

For examples, see the FitItem and FitResults objects.

OutputReport Property

Objects

Read/Write
Value: Boolean

Syntax: *FitItem object*.OutputReport

Determines whether or not the curve fit results are placed into a report.

For examples, see the FitItem and FitResults objects.

Parameter Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.Parameter(*name*)

Sets/returns the intial value of the specified parameter name for the current FitItem.

For examples, see the FitItem and FitResults objects.

ParameterCount Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.ParameterCount

Returns the number of parameters used in the regression model.

For examples, see the FitItem and FitResults objects.

ParameterDependency

ParameterDependency Method

Objects

Type: Property Get
Result Double

Syntax: *FitResults object*.ParameterDependency(*parameter name variant*)

Returns the dependency value computed for the given parameter.

For examples, see the FitItem and FitResults objects.

ParameterPValue Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ParameterPValue(*parameter variant*)

Returns the *P* value computed for the given parameter.

For examples, see the FitItem and FitResults objects.

ParameterRegressionCoefficient Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ParameterRegressionCoefficient(*parameter variant*)

Returns the value (coefficient) for the given parameter.

For examples, see the FitItem and FitResults objects.

ParametersColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.ParametersColumn

Returns/sets the output column for the parameter results.

For examples, see the FitItem and FitResults objects.

ParameterStandardError Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ParameterStandardError(*parameter variant*)

Returns the standard error computed for the given parameter.

For examples, see the FitItem and FitResults objects.

ParameterTValue Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ParameterTValue(*parameter variant*)

Returns the *t* value computed for the given parameter.

For examples, see the FitItem and FitResults objects.

PerfectFit Property

Objects

Read Only
Value: Boolean

Syntax: *FitResults object*.HasWeights

Returns whether or not the regression was a perfect fit (*R* = 1.00).

For examples, see the FitItem and FitResults objects.

Power Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.Power(*alpha*)

Returns the Power of the performed regression given an alpha value.

For examples, see the FitItem and FitResults objects.

PredictedColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/PlotEquation object*.PredictedColumn

Returns/sets the output column for the predicted dependent variable values results.

For examples, see the FitItem and FitResults objects.

PredictedValue Method

Objects

Type: Property Get
Returns: Double

Syntax: *FitResults object*.PredictedValue(*observation index long*)

Returns the predicted dependent variable value for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

PRESS Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.PRESS

Returns the PRESS statistic computed for the regression.

For examples, see the FitItem and FitResults objects.

PValue Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.PValue

Returns the *P* statistic computed for the regression.

For examples, see the FitItem and FitResults objects.

RegressionDegreesOfFreedom Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.RegressionDegreesOfFreedom

Returns the degrees of freedom (DOF) used for the regression.

For examples, see the FitItem and FitResults objects.

RegressionSumOfSquares Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.RegressionSumOfSquares

Returns the sum of squares computed for the regression.

For examples, see the FitItem and FitResults objects.

ResidualDegreesOfFreedom Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.ResidualDegreesOfFreedom

Returns the degrees of freedom of the residuals computed for the regression.

For examples, see the FitItem and FitResults objects.

ResidualsColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.ResidualsColumn

Returns/sets the output column for the residuals results.

For examples, see the FitItem and FitResults objects.

ResidualSumOfSquares Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.ResidualSumOfSquares

Returns the sum of squares of the residuals computed for the regression.

For examples, see the FitItem and FitResults objects.

ResidualValue Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.ResidualValue(*observation index long*)

Returns the residual value for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

RSquare Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.RSquare

Returns the *R* square statistic computed for the regression.

For examples, see the FitItem and FitResults objects.

Run Method

Objects

Type: Function
Result: Boolean

Syntax: *MacroItem/FitItem object*.Run

Runs a FitItem or Macro without closing the object.

Example

```
Dim Selection(3)
Selection(0) = 0
Selection(1) = 0
Selection(2) = 1
Selection(3) = &H7FFFFFF
ActiveDocument.CurrentDataItem.SelectionExtent = Selection
Dim ActiveDoc As Object
Dim CurItem As Object
Set ActiveDoc = ActiveDocument
Set CurItem = ActiveDocument.CurrentItem
Notebooks.Open(path+"\Standard.jfl")
```

```
ActiveDoc.Activate
CurItem.IsCurrentItem = True
Dim FitObject As Object
Set FitObject = Notebooks(path+"\Standard.jfl").NotebookItems("Quadratic")

FitObject.Open
FitObject.Variable("x") = "col(1)"
FitObject.Variable("y") = "col(2)"
FitObject.Run
FitObject.Finish
```

Fits a quadratic curve to the data in the first two columns of the current worksheet.

The following example shows a complete run of the fit wizard as the macro recorder records it (excluding the comments). It contains "Run" and "Finish" as well as the various statements needed to set up a curve fit session.

```
' Remember the current item and document
Dim CurItem As Object
Set CurItem = ActiveDocument.CurrentItem
Dim ActiveDoc As Object
Set ActiveDoc = ActiveDocument

' Open the fit file containing the fit we want to run.
Notebooks.Open(path+"\Standard.jfl")
Dim FitFile As Object
Set FitFile = Notebooks("C:\Data\PROJ\spw32\Standard.jfl")

' Reset the current document and worksheet to get fit data from.
ActiveDoc.Activate
CurItem.IsCurrentItem = True

' Open the fit we want to run.
Dim FitObject As Object
Set FitObject = Notebooks(path+"\Standard.jfl").NotebookItems("Single, 2 Parameter")

FitObject.Open

' Set the data format and set the variables
FitObject.DatasetType = CF_XYPAIR
FitObject.Variable("x") = "col(1)"
FitObject.Variable("y") = "col(2)"

' Run the fit. (This computes the fit results but
' does not output graphs, data, or reports.
FitObject.Run
' Set the output parameters
FitObject.OutputReport = False
FitObject.OutputEquation = False
FitObject.ResidualsColumn = -2
FitObject.PredictedColumn = -2
FitObject.ParametersColumn = -2
FitObject.OutputGraph = True
FitObject.OutputAddPlot = True
FitObject.AddPlotGraphIndex = -1
FitObject.XColumn = -1
FitObject.YColumn = -1
FitObject.ZColumn = -2

' Output the results (this would also "Run" the fit if
' we hadn't already done that.
FitObject.Finish

' Close the fit file and set the variable to "Nothing" to make sure the
' fit file is completely released. (We would not be able to reopen it
' until this is done or this macro finishes).
FitFile.Close(True)
Set FitFile = Nothing
```

RValue Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.RValue

Returns the *R* statistic computed for the regression.

For examples, see the FitItem and FitResults objects.

SetFitGraphDataRange Method

Objects

Type: Sub

Syntax: *FitItem object*.SetFitGraphDataRange(*optional xmin variant, optional xmax variant, optional ymin variant, optionalymax variant*)

Overrides default data range for curve fit graph output.

For examples, see the FitItem and FitResults objects.

SpearmanRValue Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.SpearmanRValue

Returns the Spearman *R* statistic computed for the regression.

For examples, see the FitItem and FitResults objects.

StandardErrorOfEstimate Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.StandardErrorOfEstimate

Returns the standard error of the estimate computed for the regression.

For examples, see the FitItem and FitResults objects.

StandardizedResidual Method

Objects

Type: Property Get
Result: Double

Syntax: *FitResults object*.StandardizedResidual(*observation index long*)

Returns the standardized residual value for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

StudentizedDeletedResidual Method

Objects

Type: Property Get
Returns: Double

Syntax: *FitResults object*.StudentizedDeletedResidual(*observation index long*)

Returns the Studentized deleted residual value for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

StudentizedResidual Method

Objects

Type: Property Get
Returns: Double

Syntax: *FitResults object*.StudentizedResidual(*observation index long*)

Returns the Studentized residual value for the specified independent variable index. Use the OriginalObservationIndex property to return the observation index for a given observation value.

For examples, see the FitItem and FitResults objects.

TotalDegreesOfFreedom Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.TotalDegreesOfFreedom

Returns the total degrees of freedom computed for the regression.

For examples, see the FitItem and FitResults objects.

TotalSumOfSquares Property

Objects

Read Only
Value: Double

Syntax: *FitResults object*.TotalSumOfSquares

Returns the total sum of squares computed for the regression.

For examples, see the FitItem and FitResults objects.

TrigUnit Property

Objects

Read/Write
Value: Integer

Syntax: *TransformItem/FitItem/PlotEquation object*.TrigUnit

Sets the angular unit for arguments in trigonometric functions as it is passed to the evaluator. This overrides any setting that may be contained in a transform file.

This does not read or set the trig units set for any given file, but only the default trig units used by the transform engine.

Trig Unit	Value
Radians	0
Degrees	1
Grads	2

Example

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Name = "d:\Program Files\SigmaPlot\SPW6\My Transform.xfm"

SPTransform.Open
SPTransform.TrigUnit = 0
SPTransform.Execute
SPTransform.Close(False)
```

Opens the transform file "My Transform.xfm" and runs it using radians as the trig units.

Variable Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.Variable(*name*)

Sets/returns the intial value of the specified variable name for the current FitItem (typically "x" and "y").

For examples, see the FitItem and FitResults objects.

WeightVariableName Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem object*.WeightVariableName

Sets or returns the name of the weight variable in a fit equation as a string variable. This string is empty if there is no weight variable. In the fit expression "fit f to y with weight w", the weight variable name is "w."

Example

The following example shows a complete run of the fit wizard using a weight variable.

```
Dim CurItem As Object
Set CurItem = ActiveDocument.CurrentItem
Dim ActiveDoc As Object
Set ActiveDoc = ActiveDocument
' Open the fit file containing the fit we want to run.
Notebooks.Open(path+"\Standard.jfl")
Dim FitFile As Object
Set FitFile = Notebooks(path+"\Standard.jfl")
' Reset the current document and worksheet to get fit data from.
ActiveDoc.Activate
CurItem.IsCurrentItem = True
' Open the fit we want to run.
Dim FitObject As Object
Set FitObject = Notebooks(path+"\Standard.jfl").NotebookItems("Single, 2 Parameter")

FitObject.Open
' Set the data format and set the variables
FitObject.DatasetType = CF_XYPAIR
FitObject.Variable("x") = "col(1)"
FitObject.Variable("y") = "col(2)"
' Add a Weight Variable
FitObject.Variable("WeightColumn") = "col(3)"
FitObject.WeightVariableName = "WeightColumn"
MsgBox(FitObject.WeightVariableName,0+64, "Weight Variable")
FitObject.Run
' Set the output parameters
FitObject.OutputReport = False
FitObject.OutputEquation = False
FitObject.ResidualsColumn = -2
FitObject.PredictedColumn = -2
FitObject.ParametersColumn = -2
FitObject.OutputGraph = True
FitObject.OutputAddPlot = False
FitObject.AddPlotGraphIndex = -1
FitObject.XColumn = -1
FitObject.YColumn = -1
FitObject.ZColumn = -2
FitObject.Finish

' Close the fit file and set the variable to "Nothing" to make sure the
' fit file is completely released. (We would not be able to reopen it
' until this is done or this macro finishes).
FitFile.Close(True)
Set FitFile = Nothing
```

XColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/Smoother/PlotEquation object*.XColumn

Returns/sets the output column for the x variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

YColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/Smoother/PlotEquation object*.YColumn

Returns/sets the output column for the y variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

ZColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/PlotEquation object*.ZColumn

Returns/sets the output column for the z variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

AddToGraph Property

Object

Read/Write
Value: Boolean

Syntax: *Smoother/PlotEquation object*.AddToGraph

Indicates whether the smoother or function plotter results should be plotted on the selected graph.

PlotEquation Properties and Methods

PlotEquation Properties

AddToGraph

CoordSystem

CreateGraph

CurveType

Dimension

EquationRHS

EquationSingularites

SaveOption

TrigUnit

XColumn

XEquationRHS

XIntervals

XVarName

YColumn

YEquationRHS

YIntervals

YVarName

ZColumn

PlotEquation Methods

ChangeNotebook

ClearParameters

Create

EquationLHS

Evaluate

Open

Plot

SetParameter

SetSection

SetSolverRange

Solve

XRange

YRange

AddToGraph Property

Object

Read/Write
Value: Boolean

Syntax: *Smoother/PlotEquation object*.AddToGraph

Indicates whether the smoother or function plotter results should be plotted on the selected graph.

BandWidth Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.BandWidth

Sets or returns the bandwidth method used to compute smoothed values.

ChangeNotebook Method

Objects

Type: Sub

Syntax: *PlotEquation object*.ChangeNotebook(*path variant*)

Changes the current notebook that stores the plot equation.

ClearParameters Method

Objects

Type: Sub

Syntax: *PlotEquation object*.ClearParameters

Removes all parameters and their values in computations.

CoordSystem Property

Objects

Read/Write
Value: Integer

Syntax: : *PlotEquation object*.CoordSystem

Sets or returns the coordinate system used to represent the plot

See the PlotEquation object for an example.

Create Method

Objects

Type: Sub

Syntax: *PlotEquation object*.Create(*equation string*)

Creates an equation in the current notebook.

CreateGraph Property

Object

Read/Write
Value: Boolean

Syntax: *Smoother/PlotEquation object*.CreateGraph

Indicate whether the smoother or function plotter results should be plotted on a new graph page.

CurveType Property

Objects

Read/Write
Value: Integer

Syntax: : *PlotEquation object*.CurveType

Sets or returns the equation description for the curve to be plotted.

See the PlotEquation object for an example.

Dimension Property

Objects

Read/Write
Value: Integer

Syntax: : *PlotEquation object*.Dimension

Specifies the dimension of plot.

See the PlotEquation object for an example.

EquationLHS Method

Objects

Type: Sub

Syntax: *PlotEquation object*.EquationLHS(*left hand side double*)

Specifies the constant used for the left hand side of an equation.

EquationRHS Property

Objects

Read/Write
Value: Integer

Syntax: : *PlotEquation object*.EquationRHS

Sets or returns the expression defining the plot equation.

See the PlotEquation object for an example.

EquationSingularites Property

Objects

Read Only
Value: String

Syntax: : *PlotEquation object*.EquationSingularities

Returns singularities observed when solving an equation.

Evaluate Method

Objects

Type: Function
Result: Double

Syntax: *PlotEquation object*.Evaluate(*x variable double*, *y variable variant*)

Evaluates the plot expression for the specified values of the independent variables.

Plot Method

Objects

Type: Sub

Syntax: *PlotEquation object*.Plot

Creates a plot of the current equation.

See the PlotEquation object for an example.

SaveOption Property

Objects

Read Only
Value: Boolean

Syntax: *PlotEquation object*.SaveOption

Indicate whether the equation data should be saved after plotting.

See the PlotEquation object for an example.

SetSection Method

Objects

Type: Sub

Syntax: *PlotEquation object*.SetSection(*section variant)*

Sets the notebook section containing the current equation

SetSolverRange Method

Objects

Type: Sub

Syntax: *PlotEquation object*.SetSection(*min double, max double)*

Sets the range in which to search for the solutions to an equation

Solve Method

Objects

Type: Function
Returns: Boolean

Syntax: *PlotEquation object*.Solve

Solves an equation.

TrigUnit Property

Objects

Read/Write
Value: Integer

Syntax: *TransformItem/FitItem/PlotEquation object*.TrigUnit

Sets the angular unit for arguments in trigonometric functions as it is passed to the evaluator. This overrides any setting that may be contained in a transform file.

This does not read or set the trig units set for any given file, but only the default trig units used by the transform engine.

Trig Unit	Value
Radians	0
Degrees	1
Grads	2

Example

```
Dim SPTransform As Object
Set SPTransform = ActiveDocument.NotebookItems.Add(9)
SPTransform.Name = "d:\Program Files\SigmaPlot\SPW6\My Transform.xfm"

SPTransform.Open
SPTransform.TrigUnit = 0
SPTransform.Execute
SPTransform.Close(False)
```

Opens the transform file "My Transform.xfm" and runs it using radians as the trig units.

XIntervals Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother/PlotEquation object*.XIntervals

Sets or returns the number of intervals along the x-axis for the grid of smoothing locations.

XColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/Smoother/PlotEquation object*.XColumn

Returns/sets the output column for the x variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

XEquationRHS Property

Objects

Read/Write
Value: String

Syntax: *PlotEquation object*.XEquationRHS

Sets or returns the expression defining the parametric equation for x.

See the PlotEquation object for an example.

XRange Method

Objects

Type: Sub

Syntax: *PlotEquation object*.XRange(*min double, max double)*

Sets the minimum and maximum of the range of x values

See the PlotEquation object for an example.

XVarName Property

Objects

Read/Write
Value: String

Syntax: *PlotEquation object*.XVarName

Sets or returns the name of independent variable identified with the x axis

See the PlotEquation object for an example.

YColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/Smoother/PlotEquation object*.YColumn

Returns/sets the output column for the y variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

YEquationRHS Property

Objects

Read/Write
Value: String

Syntax: *PlotEquation object*.YEquationRHS

Sets or returns the expression defining the parametric equation for y

See the PlotEquation object for an example.

YIntervals Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother/PlotEquation object*.YIntervals

Sets or returns the number of intervals along the y-axis for the grid of smoothing locations.

YRange Method

Objects

Type: Sub

Syntax: *PlotEquation object*.YRange(*min double, max double)*

Sets the minimum and maximum of the range of y values

YVarName Property

Objects

Read/Write
Value: String

Syntax: *PlotEquation object*.YVarName

Sets or returns the name of independent variable identified with the y axis

ZColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/PlotEquation object*.ZColumn

Returns/sets the output column for the z variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

Smoother Properties and Methods

Smoother Properties

AddToGraph

BandWidth

CreateGraph

Degree

OutlierRejection

PlotRawData

PredictedColumn

Type

XIntervals

XColumn

XGridColumn

XGridMax

XGridMin

XSourceColumn

YColumn

YGridColumn

YGridMax

YGridMin

YIntervals

YSourceColumn

ZColumn

ZGridColumn

ZSourceColumn

Smoother Methods

Run

SetOutputColumns

SetSourceColumns

CreateGraph Property

Object

Read/Write
Value: Boolean

Syntax: *Smoother/PlotEquation object*.CreateGraph

Indicate whether the smoother or function plotter results should be plotted on a new graph page.

Degree Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.Degree

Sets or returns the degree of the fit polynomial used in some methods.

OutlierRejection Property

Object

Read/Write
Value: Boolean

Syntax: *Smoother object*.OutlierRejection

Indicates whether the outlier-rejection algorithm should be applied.

PlotRawData Property

Object

Read/Write
Value: Boolean

Syntax: *Smoother object*. PlotRawData

Indicate whether the raw data should be plotted with the smoothing

PredictedColumn Property

Objects

Read/Write
Value: Long

Syntax: *Smoother object*.PredictedColumn

Sets or returns the output column of smoothed data values.

Proportion Property

Objects

Read/Write
Value: Double

Syntax: *Smoother object*.Proportion

Sets or returns the proportion of data used to compute each smoothed value.

ResidualColumn Property

Objects

Read/Write
Value: Long

Syntax: *Smoother object*.ResidualColumn

Sets or returns the output column of residual values.

Run Method

Objects

Type: Function
Result: Boolean

Syntax: *Smoother object*.Run

Computes the smoothed values and places the results in the worksheet and graph page.

SetOutputColumns Method

Objects

Type: Sub

Syntax: *Smoother object*.SetOutputColumns(*residual long, predicted long, x grid long, y grid long, zgrid long*)

Specifies the output data columns. All parameters must be entered. To set individual columns, use PredictedColumn.

SetParameter Method

Objects

Type: Sub

Syntax: *PlotEquation object*.SetParameter(*variable name string,variable value variant)*

Specifies the name and value of an equation parameter.

SetSourceColumns Method

Objects

Type: Sub

Syntax: *Smoother object*.SetOutputColumns(*x column long, y column long, z column long*)

Specifies the source data columns.

Type Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.Type

Sets or returns the smoothing method. Use the SPWSmoothingMethods constants to specify the type. Note that the SM_INVERSE_DISTANCE method can only be applied to 3D data.

XIntervals Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother/PlotEquation object*.XIntervals

Sets or returns the number of intervals along the x-axis for the grid of smoothing locations.

XColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/Smoother/PlotEquation object*.XColumn

Returns/sets the output column for the x variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

XgridColumn Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.XGridColumn

Sets or returns the output column of x grid coordinates.

XGridMax Property

Objects

Read/Write
Value: Double

Syntax: *Smoother object*.XGridMax

Sets or returns the maximum of the x-coordinates for the grid of smoothing locations.

XGridMin Property

Objects

Read/Write
Value: Double

Syntax: *Smoother object*.XGridMin

Sets or returns the minimum of the x-coordinates for the grid of smoothing locations.

XSourceColumn Property

Objects

Read/Write

Value: Integer

Syntax: *Smoother object*.XSourceColumn

Sets or returns the source column of x values.

YColumn Property

Objects

Read/Write
Value: Variant

Syntax: *FitItem/Smoother/PlotEquation object*.YColumn

Returns/sets the output column for the y variable values used to plot the results.

For examples, see the FitItem and FitResults objects.

YGridColumn Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.YGridColumn

Sets or returns the output column of y grid coordinates.

YGridMax Property

Objects

Read/Write
Value: Double

Syntax: *Smoother object*.YGridMax

Sets or returns the maximum of the y-coordinates for the grid of smoothing locations.

YGridMin Property

Objects

Read/Write
Value: Double

Syntax: *Smoother object*.YGridMin

Sets or returns the minimum of the y-coordinates for the grid of smoothing locations.

YIntervals Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother/PlotEquation object*.YIntervals

Sets or returns the number of intervals along the y-axis for the grid of smoothing locations.

YSourceColumn Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.YSourceColumn

Sets or returns the source column of y values.

ZGridColumn Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.ZGridColumn

Sets or returns the output column of z grid coordinates.

ZSourceColumn Property

Objects

Read/Write
Value: Integer

Syntax: *Smoother object*.ZSourceColumn

Sets or returns the source column of y values

Message Forwarding

If an object receives a message not of its own type, it does the following:

Object *Attr*	*Page*	*Graph*	*Plot*	*Axis*	*Text*	*Line*	*Symbol*	*Solid*	*Tuple*	*Function*
Page	-	(sel)	(sel)	(sel)	Label	(sel)	(sel)	(sel)	Plot	Plot
Graph	Up	-	(sel)	(sel)	Label	?	?	?	Plot	Plot
Plot	Page	Up	-	(sel)	Label	(sel)	Tuple	Tuple	Tuple	(sel)
Axis	Page	Up*	Up*	-	(sel)	(sel)	?	?	?	?
Text	?	?	?	?	-	?	?	?	?	?
Line	Page	?	?	?	?	-	?	?	?	?
Symbol	Page	?	?	?	?	(?)	-	(?)	?	?
Solid	Page	?	?	?	?	(?)	?	-	?	?
Tuple	Page	?	?	?	Label	(sel)	(sel)	(sel)	-	(sel)
Function	Page	?	?	?	(sel)	(sel)	?	?	?	-

A dash indicates that the attribute is handled locally.

(sel) indicates that the attribute is forwarded to the selected object of the type consistent with the attribute.

Label indicates the attribute is forwarded to the object's text label.

Up indicates that the attribute is forwarded to its owner. (**Up*** reflects the fact that Axes will soon be owned by graphs, not plots.)

? indicates that the attribute has no defined meaning when given to the object; results are unpredictable. **(?)** indicates possible future consideration to another object.

An object type indicates the message is forwarded to some specific object or set of objects of the type indicated. (ie, sending a Symbol attribute to a Plot will result in one or more of the Tuples to be notified.)

All objects will accept generic attributes (**SOA_**) as if they were their own (that is, they are not likely forwarded.)

SigmaPlot Constants and Enums

The following are constants that can be used within SigmaPlot instead of numeric values. Many of these are used specifically as attributes and values for the graph objects SetAttribute and GetAttribute methods.

Notebook Item Types

Colors

Function (Regression and Reference Line) Attributes

Function Options

Polynomial Linearization Operations

Group (Bag) Attributes

Object Seeking

Curve Fitter Verdicts

CurveFit DatasetTypes

Smoothing Methods

SPWPageSelectionAlignments: Alignment Constants

These are the page alignment options.

SPA_ALIGN_HLEFT	&H0000001
SPA_ALIGN_HCENTER	&H0000002
SPA_ALIGN_HRIGHT	&H0000003
SPA_ALIGN_VTOP	&H0000010
SPA_ALIGN_VCENTER	&H0000020
SPA_ALIGN_VBOTTOM	&H0000030

SPWGraphAttribute: Axis Attributes

Axis options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SAA_BASE	&H00000400	
SAA_END	&H000004FF	
SAA_NAME	&H00000400	The name of the axis; this is also the axis title
SAA_TYPE	&H00000401	The axis scale type. Use SAA_TYPE constants
SAA_DIM	&H00000402	The dimension the axis occupies. Use DIM constants
SAA_OPTIONS	&H00000403	Options to apply to the axis as a whole. Use SAA_FLAG constants
SAA_FROMVAL	&H00000406	Determines one of the extremes of the axis range (only used if auto-scaling is off)
SAA_TOVAL	&H00000407	Determines the other extreme of the axis range (only used if auto-scaling is off)
SAA_ORGVAL	&H00000408	Determines the tick origin value (only meaningful for linear scales)
SAA_INTVAL	&H00000409	Determines the tick interval value. This attribute varies depending on the axis

		scale type (Use SAA_DENS_ constants)
SAA_SELECTLINE	&H0000040A	Determines which specific line is to be influenced by subsequent attribute messages (i.e., sets the Selected Line). For example, some attributes might expect SAA_LINE_MAJOR or SAA_LINE_MINOR to be selected. Use SAA_LINE constants
SAA_HLINE	&H0000040B	Assigns the line object for the Selected Line
SAA_CUSTOMCOL	&H0000040C	If custom mapping scale is selected, this is the column that represents the interpolation points for that axis
SAA_SUB1OPTIONS	&H00000410	Options to apply to the first of two (or four) subaxes. Use SAA_SUB constants
SAA_SUB2OPTIONS	&H00000411	Options to apply to the second of two (or four) subaxes. Use SAA_SUB constants
SAA_POS1	&H00000414	The displaced position of the first of two (or four) axes, as expressed from 'normal' position of 0. Positive numbers represents to right/above normal
SAA_POS2	&H00000415	The displaced position of the second of two (or four) axes, as expressed from 'normal' position of 0. Positive numbers represents to right/above normal
SAA_POS1PERMILL	&H00000418	Duplicates SAA_POS1, but expresses the units in percentage form (using tenths of percents)
SAA_POS2PERMILL	&H00000419	Duplicates SAA_POS2, but expresses the units in percentage form (using tenths of percents)
SAA_TICLABELCOLUSED	&H00000420	
SAA_HTEXT	&H00000421	
SAA_TICLABEL	&H00000422	Assigns the axis tick label. This label is never actually shown directly, but elements are accessed in order to create the tick labels. This attribute is influenced by SAA_LINE; the SAA_LINE_MAJOR or SAA_LINE_MINOR line should be selected before using this attribute
SAA_TICLABELCOL	&H00000423	Defines the column where tick labels are to come from. This attribute is influenced by SAA_LINE
SAA_TICLABELNOTATION	&H00000424	Determines the format that tick labels are presented in. Use SAA_TLBL Constants . This attribute is influenced by SAA_LINE
SAA_TICLABELTHRESHMIN	&H00000425	Determines the smallest log magnitude between which logarithmic labels will not be produced
SAA_TICLABELTHRESHMAX	&H00000426	Determines the largest log magnitude between which logarithmic labels will not be produced
SAA_TICLABELOFFSET	&H00000427	The distance the tick labels are displaced from the axis line (only implemented as a get)
SAA_TICLABELALIGN	&H00000428	The alignment method to be used for the tick labels. Use SAA_ALIGN constants
SAA_TICLABELPREFIX	&H00000429	A prefix to be placed before each (major or minor) tick label. This attribute is influenced by SAA_LINE
SAA_TICLABELSUFFIX	&H0000042A	A suffix to be placed after each (major or minor) tick label. This attribute is influenced by SAA_LINE
SAA_TICLABELPLACES	&H0000042B	The number of places of precision to present tick labels with
SAA_TICLABELAUTOPREC	&H0000042C	Indicates whether ticks label precision should be automatically determined
SAA_TICLABELFACTOR	&H0000042D	A factor to remove from tick mark values prior to producing a string. Permits factoring out of powers of ten, for example. This attribute is influenced by SAA_LINE
SAA_TICSIZE	&H0000042E	Determines the tick size. This attribute is influenced by SAA_LINE
SAA_MINORFREQ	&H0000042F	Determines the minor tick frequency, if applicable. This attribute varies depending on the axis scale type. For Log ticks, use the SAA_LOGTIC constants
SAA_TICCOL	&H00000430	Determines the column from which ticks (as distinguished from tick label values) will be taken. Depends on SAA_TICKCOLUSED
SAA_TICCOLUSED	&H00000431	Determines if SAA_TICKCOL is used or not; otherwise automatic tick generation will take place
SAA_TICCOLIDENTMAJOR	&H00000432	Determines the column from which tick-identity is gotten (is the tick major or minor). This depends both on SAA_TICCOLUSED and SAA_TICKCOLIDENTUSED

SAA_TICCOLIDENTUSED	&H00000433	Determines if the attribute SAA_TICCOLIDENTMAJOR is used. This attribute is ignored if SAA_TICKCOLUSED is false
SAA_BREAKMIN	&H00000434	The lower bound of the break. This attribute depends on SAA_BREAKON
SAA_BREAKMAX	&H00000435	The upper bound of the break. This attribute depends on SAA_BREAKON
SAA_BREAKPOS	&H00000436	Determines the position of the break as a percentage of the axis along which the break is to be placed. This attribute depends on SAA_BREAKON
SAA_BREAKON	&H00000437	Determines whether the break is visible or not
SAA_BREAKPOSTINT	&H00000438	Determines the post-break interval (if applicable)
SAA_BREAKPOSTORG	&H00000439	Determines the post-break origin
SAA_BREAKTYPE	&H0000043A	Determines the break symbol. Use SAA_BREAK constants
SAA_BREAKGAP	&H0000043B	Determines the gap size of the break: the distance between the two axis lines
SAA_BREAKWIDTH	&H0000043C	Determines the width of the break symbol
SAA_AUTOFROMVAL	&H00000440	Determines the most recent 'from' value as calculated by the validation routines. This is a Get-Only attribute
SAA_AUTOTOVAL	&H00000441	Determines the most recent 'to' value as calculated by the validation routines. This is a Get-Only attribute
SAA_AUTOINCRVAL	&H00000442	Determines the most recent incremement value (as appropriate, determined by axis scale type) as calculated by the validation routines
SAA_SHOWNAME	&H00000450	
SAA_ENUMPLOTSUSING	&H00000451	
SAA_MAJORFREQINDIRECT	&H00000452	
SAA_HNAME	&H00000453	
SAA_SELECTTIC	&H00000454	Use the SAA_TIC constants
SAA_OLDSTYLEDATELABEL	&H00000455	
SAA_OLDSTYLEDATELABELON	&H00000456	
SAA_OLDSTYLEDATELABELFROM	&H00000457	
SAA_OLDSTYLEDATELABELTO	&H00000458	
SAA_OLDSTYLEDATELABELBY	&H00000459	
SAA_OLDSTYLEDATELABELGO	&H0000045a	
SAA_OLDSTYLEDATELABELLEN	&H0000045b	
SAA_VALIDMAXVAL	&H00000460	
SAA_VALIDMINVAL	&H00000461	
SAA_VALIDINTVAL	&H00000462	
SAA_VALIDORGVAL	&H00000463	
SAA_HNAME2	&H00000464	
SAA_SUB1FRAMEREF	&H00000465	
SAA_SUB2FRAMEREF	&H00000466	

SAA_POLARAXISSTART	&H00000467	
SAA_POS3	&H00000468	
SAA_POS4	&H00000469	
SAA_SUB3OPTIONS	&H0000046a	
SAA_SUB4OPTIONS	&H0000046b	
SAA_POS3PERMILL	&H0000046c	
SAA_POS4PERMILL	&H0000046d	
SAA_POLARPERIOD	&H0000046e	
SAA_POLARUNITS	&H0000046f	
SAA_HNAME3	&H00000470	
SAA_HNAME4	&H00000471	
SAA_MAJORFREQPROBABILITY	&H00000472	
SAA_TICLABELDATEFORM	&H00000473	
SAA_TICLABELTIMEFORM	&H00000474	
SAA_TICLABELDTIUNIT	&H00000475	Use SAA_DTUNIT constants
SAA_TICLABELDTIUNITCOUNT	&H00000476	
SAA_TICLABELDTIUNITVALID	&H00000477	
SAA_TICLABELDTIUNITCOUNTVALID	&H00000478	
SAA_TRANSFORMVALUE	&H00000479	
SAA_UNTRANSFORMVALUE	&H0000047a	
SAA_HPRIMARYNAME	&H0000047B	
SAA_RTFNAME	&H0000047C	
SAA_RTFNAME1	&H0000047D	
SAA_RTFNAME2	&H0000047E	
SAA_RTFNAME3	&H0000047F	
SAA_RTFNAME4	&H00000480	

SPWAxisWizardAxisPosition: Axis Wizard Axis Position Constants

Add Axis wizard new axis location options.

AxisPosRightNormal	0	Right side of graph at 0%
AxisPosRightOffset	1.	Right side of graph offset 20%
AxisPosTopNormal	2.	Top of graph at 0%

AxisPosTopOffset	3.	Top of graph offset 20%
AxisPosLeftNormal	4.	Left side of graph at 0%
AxisPosLeftOffset	5.	Left side of graph offset 20%
AxisPosBottomNormal	6.	Bottom of graph at 0%
AxisPosBottomOffset	7.	Bottom of graph offset 20%

SPWStockSchemes: Built-in Scheme Constants

STOCKSCHEME_COLOR_BW	&H00010001
STOCKSCHEME_COLOR_GRAYS	&H00020001
STOCKSCHEME_COLOR_EARTH	&H00030001
STOCKSCHEME_COLOR_FOREST	&H00040001
STOCKSCHEME_COLOR_OCEAN	&H00050001
STOCKSCHEME_COLOR_RAINBOW	&H00060001
STOCKSCHEME_COLOR_OLDINCREMENT	&H00070001
STOCKSCHEME_SYMBOL_DOUBLE	&H00010002
STOCKSCHEME_SYMBOL_MONOCHROME	&H00020002
STOCKSCHEME_SYMBOL_DOTTEDDOUBLE	&H00030002
STOCKSCHEME_SYMBOL_OLDINCREMENT	&H00040002
STOCKSCHEME_LINE_MONOCHROME	&H00010003
STOCKSCHEME_LINE_OLDINCREMENT	&H00020003
STOCKSCHEME_PATTERN_MONOCHROME	&H00010004
STOCKSCHEME_PATTERN_OLDINCREMENT	&H00020004

SPWColorValues : Color Constants

These colors correspond to the built-in colors and scheme colors.

RGB_EMPTY	&HFF000000	None (no color)
RGB_BLACK	&H00000000	rgb(0,0,0)
RGB_RED	&H000000FF	rgb(255,0,0)
RGB_GREEN	&H0000FF00	rgb(0,255,0)
RGB_YELLOW	&H0000FFFF	rgb(255,255,0)
RGB_BLUE	&H00FF0000	rgb(0,0,255)
RGB_PINK	&H00FF00FF	rgb(255,0,255)

RGB_CYAN	&H00FFFF00	rgb(0,255,255)
RGB_WHITE	&H00FFFFFF	rgb(255,255,255)
RGB_DKGRAY	&H00808080	rgb(128,128,128)
RGB_DKRED	&H00000080	rgb(128,0,0)
RGB_DKGREEN	&H00008000	rgb(0,128,0)
RGB_DKYELLOW	&H00008080	rgb(128,128,0)
RGB_DKBLUE	&H00800000	rgb(0,0,128)
RGB_DKPINK	&H00800080	rgb(128,0,128)
RGB_DKCYAN	&H00808000	rgb(0,128,128)
RGB_GRAY	&H00C0C0C0	rgb(192,192,192)
RGB_GRAYS1	RGB_BLACK	rgb(0,0,0)
RGB_GRAYS2	RGB_GRAY	rgb(192,192,192)
RGB_GRAYS3	&H00606060	rgb(64,64,64)
RGB_GRAYS4	&H00E0E0E0	rgb(224,224,224)
RGB_GRAYS5	&H00404040	rgb(32,32,32)
RGB_GRAYS6	&H00808080	rgb(128,128,128)
RGB_EARTHTONES1	&H00800000	rgb(128,0,0)
RGB_EARTHTONES2	&H00B6C000	rgb(192,192,0)
RGB_EARTHTONES3	&H00600000	rgb(96,0,0)
RGB_EARTHTONES4	&H00FF8000	rgb(255,128,0)
RGB_EARTHTONES5	&H00804000	rgb(128,64,0)
RGB_EARTHTONES6	&H00808000	rgb(128,128,0)
RGB_OCEAN1	&H00000080	rgb(0,0,128)
RGB_OCEAN2	&H000080FF	rgb(0,128,255)
RGB_OCEAN3	&H00000060	rgb(0,0,96)
RGB_OCEAN4	&H00008080	rgb(0,128,128)
RGB_OCEAN5	&H00004080	rgb(0,64,128)
RGB_OCEAN6	&H0000E0E0	rgb(0,224,224)
RGB_FOREST1	&H00004000	rgb(0,64,0)
RGB_FOREST2	&H0000FF00	rgb(0,255,0)
RGB_FOREST3	&H00008000	rgb(0,128,0)
RGB_FOREST4	&H00C0FF00	rgb(192,255,0)
RGB_FOREST5	&H0040C000	rgb(64,192,0)

RGB_FOREST6	&H00FFFF00	rgb(255,255,0)
RGB_MUTEDRAINBOW1	&H00800000	rgb(128,0,0)
RGB_MUTEDRAINBOW2	&H00FF8000	rgb(255,128,0)
RGB_MUTEDRAINBOW3	&H00C0C000	rgb(192,192,0)
RGB_MUTEDRAINBOW4	&H00008000	rgb(0,128,0)
RGB_MUTEDRAINBOW5	&H00008080	rgb(0,128,128)
RGB_MUTEDRAINBOW6	&H00004080	rgb(0,64,128)
RGB_MUTEDRAINBOW7	&H00800080	rgb(128,0,128)

CFFitVerdict: Curve Fitter Verdicts Constants

These correspond to the status and error messages received in the Regression Wizard curve fit results panel. These are useful if you want to filter results before presenting them.

CFV_ZEROITERATIONS	0	No attempt to fit because iterations set to 0.
CFV_TOOMANYITERATIONS	1.	Did not converge, exceeded maximum number of iterations.
CFV_INNERLOOPFAILURE	2.	Did not converge, inner loop failure.
CFV_CONVERGENCE	3.	Converged, tolerance satisfied.
CFV_NOCHANGECONVERGENCE	4.	Converged, zero parameter changes.
CFV_NOPATIENCE	5.	Terminated by user.
CFV_DISASTER	6.	This condition will almost always produce a crash if the Finish method is called. Use this verdict to trap numerical crashes.
CFV_OVERFLOW	7.	This condition is likely to produce a crash if the Finish method is called. Use this verdict to trap numerical crashes.
CFV_NULL	8.	
Secondary and tertiary results		
CFV_TRYDIFFERENTPARAMETERS	9.	No parameters to fit.
CFV_ARRAYILLCONDITIONED	1.	Parameters may not be valid. Array ill conditioned on final iteration.
CFV_ARRAYSINGULAR	1.	Parameters may not be valid. Array numerically singular on final iteration.
CFV_PARTIALDERIVATIVEOVERFLOW	1.	Parameters may not be valid. Overflow in partial derivatives.
CFV_FLAKYCONSTRAINTS	1.	Bad constraint.
CFV_UNEXPECTEDRESULT	1.	

CurveFitDatasetTypes: Curve Fitter Dataset Type Constants

These constants represent the data format options available from the Regression Wizard.

CF_XYPAIR	0	XY Pair
CF_XYZTRIPLET	1.	XYZ
CF_FROMCODE	2.	From Code
CF_SINGLEY	3.	Y only
CF_XMANYY	4.	XY col means
CF_MANYY	5.	Y col means only
CF_XYMANYZ	6.	XY Many Z
CF_MANYZ	7.	Many Z
CF_XMANYINDEPENDENT	8.	X many Independent
CF_XREPY	9.	X Y Replicate

CF_REPY	1.	Y Replicates
CF_XYREPZ	1.	X Y Z Replicates
CF_LASTTYPE	CF_XYREPZ	

SPWDimensionID: Dimension Constants

Direction/dimension values. These specify the axis or direction being operated upon.

DIM_NONE	0	No dimension
DIM_MIN	1.	Toward minimum ('down')
DIM_X	1.	The X dimension
DIM_Y	2.	The Y dimension
DIM_Z	3.	The Z dimension (if applicable)
DIM_MAX	3	Toward maximum ('up')

Finding Constant Values

Recorded macros often display constants for specific item properties. SigmPlot usually represents these values in a hexadecimal format (&H########). To find more information about a specific constant, use the full-text search ability of Automation Help.

- Open Automation Help.
- Click the Find tab of the Help Topics dialog box.
- If prompted, build the find database with maximum search capabilities.
- Click the Options button.
- Set the Find Options to show words that contain the characters you type.
- Click OK.
- Copy and paste (or type) the constant value into the Find dialog.
- Select any desired matching words from the generated list.
- Select and display the desired topic from the final list.

SPWGraphAttribute: Function Attributes

Reference line options.

SFA_BASE	&H00000A00	
SFA_END	&H00000AFF	
SFA_OPTIONS	&H00000A01	Options to control the appearance and behavior of functions. Use SFA_FLAG constants
SFA_ORDER	&H00000A02	The order of the function polynomial

SFA_PREOP	&H00000A03	One or more operations to be performed on the independent values before applying them to the polynomial. Use the SFA_OP constants
SFA_POSTOP	&H00000A04	One or more operations to be performed on the result of the polynomial, before producing the dependant value. Use the SFA_OP constants
SFA_PREOPCONST	&H00000A05	Determines the constant used in any PREOP *Multiplication By A Constant*
SFA_POSTOPCONST	&H00000A06	Determines the constant used in any POSTOP *Multiplication By A Constant*
SFA_SELECTTERMS	&H00000A07	A MAKELONG of the first and last polynomial term to select for access. MAKELONG(0,0) selects the constant term; MAKELONG(1,1) selects the first order term, MAKELONG (0,1) selects both of these
SFA_TERMS	&H00000A08	Defines the values used in the selected terms (as determined by SFA_SELECTTERMS, which this attribute depends on)
SFA_FROM	&H00000A09	The first value that determines the extent of the functions domain. The function object will map values from the domain of the polynomial onto the range, and produce the curve that results
SFA_TO	&H00000A0A	The second value that determines the extent of the functions domain. The function object will map values from the domain of the polynomial onto the range, and produce the curve that results
SFA_RESOLUTION	&H00000A0B	The number of steps to take along the domain
SFA_COLUMN	&H00000A0C	The column to take values from (in lieu of using the range generated from SFA_FROM and SFA_TO) which apply to the function polynomial. This attribute depends on SFA_OPTIONS to have SFA_FLAG_FROMCOL set
SFA_HLINE	&H00000A0D	The line object used in rendering the function
SFA_AUGMENTFUNC	&H00000A0E	A function by which this function's range is augmented. Permits meta-functions of the form f(*x*) + g(*x*)
SFA_COMPUTE	&H00000A0F	(Read Only) The value referred to is mapped from the function's domain to its range, and the results placed back into the number
SFA_RANGEMIN	&H00000A10	(Read Only) The minimum value of the range produced within the domain of the function
SFA_RANGEMAX	&H00000A11	(Read Only) The maximum value of the range produced within the domain of the function
SFA_LABELA	&H00000A12	
SFA_LABELB	&H00000A13	
SFA_EXTRANUM	&H00000A13	
SFA_HLABELA	&H00000A14	
SFA_HLABELB	&H00000A15	

SPWGraphAttribute: General Object Attributes

General object options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SOA_POS	&H00000001	Position. This is the base point of the graph, and its definition varies with the coordinate system used.
SOA_EXTSHAPE	&H00000003	Extent shape. Use the SOA_EXT constants
SOA_COLOR	&H00000004	This is the color of the fill (background) of the object
SOA_RESET	&H00000005	An instruction to reset to default values
SOA_COPYTO	&H00000006	An instruction to copy all attributes from another object
SOA_LEFT	&H00000007	Left position (in 1000th in)

SOA_TOP	&H00000008	Top position (in 1000th in)
SOA_RIGHT	&H00000009	(obsolete)
SOA_BOTTOM	&H0000000A	(obsolete)
SOA_EXTENT	&H0000000B	
SOA_HITTEST	&H0000000C	
SOA_OFFSET	&H0000000D	
SOA_EXTPOLYSIZE	&H0000000E	
SOA_EXTPOLYPOINTS	&H0000000F	
SOA_ENUMATTRS	&H00000010	
SOA_TYPE	&H00000011	
SOA_HITTESTRECT	&H00000012	
SOA_EXTENTAREA	&H00000013	
SOA_FANCYEXTENT	&H00000014	
SOA_HITTESTGROSS	&H00000015	
SOA_RENDEREXTENT	&H00000016	
SOA_ALIGNHLEFT	&H00000017	
SOA_ALIGNHCENTER	&H00000018	
SOA_ALIGNHRIGHT	&H00000019	
SOA_ALIGNVTOP	&H0000001A	
SOA_ALIGNVCENTER	&H0000001B	
SOA_ALIGNVBOTTOM	&H0000001C	
SOA_ENUMDATA	&H0000001D	
SOA_ENUMCHILDREN	&H0000001e	Tells object to enumerate children
SOA_INVALIDATERECT	&H0000001f	Forces the current rect to be marked invalid
SOA_COMPACT	&H00000020	
SOA_SIZEEX	&H00000021	Requires array of bottom and right sizes (using absolute coordinate system). Generated when recording a size operation.
SOA_RESIZECOMPONENTS	&H00000022	
SOA_POSEX	&H00000023	Requires array of top and left positions (using absolute coordinate system). Generated when recording a postion operation.
SOA_HITTESTRESULT	&H00000024	Get and set attribute
SOA_ISOBJECTVALIDFORVERSION	&H00000025	
SOA_TRANSLATETOVERSIONSET	&H00000026	
SOA_NAME	&H00000027	Object name string

SOA_VERSIONSET	&H00000028	
SOA_OWNER	&H00000029	Gets an object's owner
SOA_DOCOWNER	&H0000002A	Gets object's document
SOA_QUERYSAVEOBJECT	&H0000002B	Returns TRUE if object needs to be saved
SOA_VALIDATE	&H0000009E	
SOA_INVALIDATE	&H0000009F	

SPWGraphAttribute: Graph Attribute Constants

Graph attribute settings. These encompass the following constants types:

SAA Axis Attributes

SBA Group (Bag) Atrtibutes

SDA Solid Attributes

SEA Line Attributes

SFA Function Attributes

SGA Graph Attributes

SLA Plot Attributes

SNA Tuple Attributes

SOA General Object Attributes

SPA Page Attributes

SSA Symbol Attributes

STA Text Attributes

SPWGraphAttribute: Graph Attributes

Graph options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SGA_BASE	&H00000200	
SGA_END	&H000002FF	
SGA_NAME	&H00000200	The name of the graph, also used for the graph title
SGA_COORDSYSTEM	&H00000201	Determines the coordinate system to be used by the graph—use SGA_COORD constants to set
SGA_ADDPLOT	&H00000202	Adds a new plot to the graph; it becomes the Current Plot

SGA_DELPLOT	&H00000203	Deletes the Current Plot
SGA_HORANGLE	&H00000204	3D horizontal rotaion
SGA_ELANGLE	&H00000205	3D vertical (elevation) angle
SGA_PERSPECTIVE	&H00000206	3D perspective
SGA_SHOWNAME	&H00000207	Display the graph title
SGA_ORGTYPE	&H00000209	
SGA_SELECTLINE	&H0000020A	Use SGA_LINE constants
SGA_HLINE	&H0000020B	
SGA_ENUMPLOTS	&H0000020C	
SGA_ENUMAXES	&H0000020D	
SGA_ADDAXIS	&H0000020E	Adds a new axis
SGA_DELAXIS	&H0000020F	Deletes the current axis
SGA_PLANECOLORXYBACK	&H00000210	Color for XY backplane
SGA_PLANECOLORXYFRONT	&H00000211	
SGA_PLANECOLORYZBACK	&H00000212	Color for YZ backplane
SGA_PLANECOLORYZFRONT	&H00000213	
SGA_PLANECOLORZXBACK	&H00000214	Color for ZX backplane
SGA_PLANECOLORZXFRONT	&H00000215	
SGA_PLANESTYLEXYBACK	&H00000216	
SGA_PLANESTYLEXYFRONT	&H00000217	
SGA_PLANESTYLEYZBACK	&H00000218	
SGA_PLANESTYLEYZFRONT	&H00000219	
SGA_PLANESTYLEZXBACK	&H0000021A	
SGA_PLANESTYLEZXFRONT	&H0000021B	
SGA_PLANEHATCHXYBACK	&H0000021C	Pattern for XY backplane
SGA_PLANEHATCHXYFRONT	&H0000021D	
SGA_PLANEHATCHYZBACK	&H0000021E	Pattern for YZ backplane
SGA_PLANEHATCHYZFRONT	&H0000021F	
SGA_PLANEHATCHZXBACK	&H00000220	Pattern for ZX backplane
SGA_PLANEHATCHZXFRONT	&H00000221	
SGA_PLANEXYBACK	&H00000230	

SGA_PLANEYZBACK	&H00000231	
SGA_PLANEZXBACK	&H00000232	
SGA_FRAMEORG	&H00000240	
SGA_FRAMENONORG	&H00000241	
SGA_FRAMENEAR	&H00000242	
SGA_FRAMEFAR	&H00000243	
SGA_MAXDIM	&H00000244	
SGA_AXESTOFRONT	&H00000245	Moves 3D axes to front
SGA_CREATEPLOT	&H00000246	
SGA_NTHPLOT	&H00000247	
SGA_NTHAXIS	&H00000248	
SGA_PLOTBYNAME	&H00000249	
SGA_AXISBYNAME	&H0000024A	
SGA_CURRENTPLOT	&H0000024B	
SGA_CURRENTAXIS	&H0000024C	
SGA_RENDERED	&H0000024D	
SGA_PLOTBYHANDLE	&H0000024E	
SGA_AXISBYHANDLE	&H0000024F	
SGA_REMOVEAXIS	&H00000250	
SGA_REMOVEPLOT	&H00000251	
SGA_HNAME	&H00000252	
SGA_HAUTOLEGENDBAG	&H00000253	
SGA_NTHAUTOLEGEND	&H00000254	
SGA_FLAGS	&H00000255	Use SGA_FLAGS constants
SGA_AUTOLEGENDLINESPACING	&H00000256	
SGA_AUTOLEGENDCOLSPACING	&H00000257	
SGA_AUTOLEGENDMOVED	&H00000258	
SGA_NUMLEGENDSCHANGED	&H00000259	
SGA_AUTOLEGENDSHOW	&H00000260	
SGA_CREATENEXTAUTOLEGEND	&H00000261	
SGA_HAUTOLEGENDSOLID	&H00000262	
SGA_3DLIGHTCOLOR	&H00000263	

SGA_3DLIGHTHORANGLE	&H00000264
SGA_3DLIGHTELANGLE	&H00000265
SGA_SELECTPLOT	&H00000266
SGA_TERNARYTOTAL	&H00000267
SGA_TERNARYTRIANGLEEXTENT	&H00000268
SGA_CURRENTLEGENDTEXT	&H00000269
SGA_CURRENTLEGENDSTYLE	&H0000026A
SGA_CURRENTLEGENDOPTIONS	&H0000026B

SPWGraphAttribute: Group (Bag) Attributes

Grouped object options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SBA_BASE	&H00000C00	
SBA_END	&H00000CFF	
SBA_OPTIONS	&H00000C01	Gets/Sets options for the bag. There are no flags currently defined for this message
SBA_SEEK	&H00000C02	Moves the bag's current index pointer to point to the object's Position in its list. Position is an ordinal index. See the predefined SBA_SEEK constants (Set Only)
SBA_NEXT	&H00000C03	Stores specified Object into the current list element and advances the pointer to the next element. Does not free the existing object handle, which is overwritten, so use with caution (Set Only)
SBA_SEARCH	&H00000C04	Searches the bag's list for the specified object, returning its index in the list if found, or -1 if not. The bag's current object index is left pointing to the located object, if found (Set Only)
SBA_HLIST	&H00000C05	Returns the handle of the bag's object list (Get Only)
SBA_RESET	&H00000C06	Same as SBA_SEEK(SBA_SEEK_REWIND), but a little more efficient (Set Only)
SBA_THIS	&H00000C07	Stores specified Object into the current list element. Does not free the existing object handle, which is overwritten, so use with caution (Set Only)
SBA_APPEND	&H00000C08	Appends specified Object to the bag's list of objects. This also apprises the page of the object, and marks the object selected before putting it in the bag (Set only)
SBA_ENUMBAGOBJECTS	&H00000c09	Enumerates the objects in the bag, calling for each object therein (Set Only)
SBA_OBJECTBYHANDLE	&H00000c0a	Returns the index of the specified object within the bag's list, or -1 if not found (Get Only)
SBA_NTHGPOBJECT	&H00000c0b	Returns the handle of the specified index in the bag, or NULL, if not found (Get Only)
SBA_DELETEOBJECT	&H00000c0c	Finds the specified object in the bag, deletes it, and removes it from the bag's list (Set Only)
SBA_REMOVEOBJECT	&H00000c0d	Finds the specified object in the bag, and removes it from the bag's list, but leaves the object undeleted. Presumably, another reference to the object exists and is later used to delete it, thereby preventing memory corruption. (Set Only)
SBA_TYPE	&H00000c0e	
SBA_HAUTOLEGENDGRAPH	&H00000c0f	
SBA_INSERT	&H00000c10	

SPWDeleteInsertDirection: Insert/Delete Direction Constants

Set the insert/delete direction for worksheet cells

DeleteUp	0	Shift cells up
InsertDown	1.	Shift cells down
DeleteLeft	2.	Shift cells left
InsertRight	3.	Shift cells right

SPWGraphAttribute: Line Attributes

Line options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SEA_BASE	&H00000600	
SEA_END	&H000006FF	
SEA_THICKNESS	&H00000601	The thickness of the line
SEA_LINETYPE	&H00000602	The type of line style. Use SEA_LINE constants
SEA_LINEEND1	&H00000603	A MAKELONG consisting of an SEA_END constant, and a parameter number, describing the treatment for the first endpoint. (The number might represent the angle of an arrow-head, for example.) Use SEA_END constants
SEA_LINEEND2	&H00000604	A MAKELONG consisting of an SEA_END constant, and a parameter number, describing the treatment for the second endpoint. (The number might represent the angle of an arrow-head, for example.) Use SEA_END constants
SEA_ENDSIZE	&H00000605	The size of the endpoint treatment. Depends on SEA_LINEEND1 and SEA_LINEEND2
SEA_COLOR	&H00000606	Sets the line color. Same as SOA_COLOR
SEA_END1TYPE	&H00000607	Set the shape for the beginning of the line. Use the SEA_END constants
SEA_END2TYPE	&H00000608	Set the shape for the end of the line. Use the SEA_END constants
SEA_END1ANGLE	&H00000609	Set the arrowhead angle for the arrowhead at the start of a line, in 10ths of a degree
SEA_END2ANGLE	&H0000060A	Set the arrowhead angle for the arrowhead at the end of a line, in 10ths of a degree
SEA_END1POINT	&H0000060B	
SEA_END2POINT	&H0000060C	
SEA_SEGMENTS	&H0000060D	
SEA_LPPOINTS	&H0000060E	
SEA_OPTIONS	&H0000060F	
SEA_END1SIZE	&H00000610	Set the size of the symbol at the start of a line, in 1000th of an inch
SEA_END2SIZE	&H00000611	Set the size of the symbol at the end of a line, in 1000th of an inch
SEA_TYPEREPEAT	&H00000612	Sets the STOCKSCHEME used for the line type

SEA_COLORREPEAT	&H00000613	Sets the STOCKSCHEME fill color used for the line color
SEA_TYPECOL	&H00000614	Sets the source column used for line type
SEA_COLORCOL	&H00000615	Sets the source column used for line color

SPWNotebookComponentType: Notebook Item Type Constants

CT_WORKSHEET	1.
CT_GRAPHICPAGE	2.
CT_FOLDER	3.
CT_STATTEST	4.
CT_REPORT	5.
CT_FIT	6.
CT_NOTEBOOK	7.
CT_EXCELWORKSHEET	8.
CT_TRANSFORM	9.
CT_MACRO	1.
CT_NUMBEROFTYPES	1.

SPWGraphicObjectType: Object Constants

Graphic object types.

GPT_OBJECT	0
GPT_PAGE	1.
GPT_GRAPH	2.
GPT_PLOT	3.
GPT_AXIS	4.
GPT_TEXT	5.
GPT_LINE	6.
GPT_SYMBOL	7.
GPT_SOLID	8.
GPT_TUPLE	9.
GPT_FUNCTION	1.
GPT_EXTERNAL	1.
GPT_BAG	1.
GPT_DOCUMENT	1.
GPT_DATATABLE	1.

SPWOptionFlagControlBits: Option Flag Control Bit Constants

Constants used in the FlagOn/FlagOff functions embedded in all SigmaPlot macros

FLAG_SET_BIT	1.
FLAG_CLEAR_BIT	2.

SPWGraphAttribute: Page Attributes

Graph Page options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SPA_BASE	&H00000100	
SPA_END	&H000001FF	
SPA_NAME	&H00000100	Page name
SPA_WIDTH	&H00000101	Page width
SPA_HEIGHT	&H00000102	Page height
SPA_LEFTMARGIN	&H00000103	Left margin
SPA_TOPMARGIN	&H00000104	Top margin
SPA_RIGHTMARGIN	&H00000105	Right margin
SPA_BOTTOMMARGIN	&H00000106	Bottom margin
SPA_OPTIONS	&H00000109	
SPA_RENDERMETHOD	&H0000010A	
SPA_ADDOBJECT	&H0000010B	
SPA_DELOBJECT	&H0000010C	
SPA_FRONTMOST	&H0000010D	
SPA_BACKMOST	&H0000010E	
SPA_RENDERRESULT	&H00000111	
SPA_CURRENTOBJECT	&H00000112	
SPA_DEFAULTDW	&H00000113	
SPA_COLNOTIFY	&H00000114	
SPA_CREATEGRAPH	&H00000115	
SPA_ENUMGRAPHS	&H00000116	
SPA_ENUMSELECTIONEX	&H00000117	
SPA_ADDTOSELECTION	&H00000118	
SPA_CLEARSELECT	&H00000119	
SPA_SELECTOBJECT	&H0000011A	
SPA_NTHGRAPH	&H0000011B	
SPA_GRAPHBYNAME	&H0000011C	
SPA_NEXTGRAPHRECT	&H0000011D	
SPA_NEXTGRAPHXOFF	&H0000011E	
SPA_NEXTGRAPHYOFF	&H0000011F	
SPA_GROUP	&H00000120	

SPA_UNGROUP	&H00000121	
SPA_BRINGTOFRONT	&H00000122	Bring to front
SPA_SENDTOBACK	&H00000123	Send to back
SPA_ENUMOBJECTS	&H00000125	
SPA_GRAPHBYHANDLE	&H00000126	
SPA_CLEARPAGE	&H00000127	
SPA_REMOVEOBJECT	&H00000128	
SPA_REMOVEGRAPH	&H00000129	
SPA_DELGRAPH	&H0000012A	
SPA_ALIGNSELECTIONS	&H0000012B	Object alignment options; use Alignment Constants
SPA_POSITIONSELECTIONS	&H0000012C	
SPA_HNAME	&H0000012D	
SPA_CREATEOBJECTFROM	&H0000012E	
SPA_NUMSELECTIONS	&H0000012F	
SPA_HASCHANGED	&H00000130	
SPA_NTHGPOBJECT	&H00000131	
SPA_OBJECTBYHANDLE	&H00000132	
SPA_RENDERQUALITY	&H00000133	
SPA_SELECTALL	&H00000134	Select all objects on page
SPA_FORCEUPDATE	&H00000135	Force page redraw
SPA_COLTITLENOTIFY	&H00000136	Notify legend of column title change
SPA_DLLVERSION	&H00000137	
SPA_VERSIONSET	&H00000138	
SPA_DEFAULTHDATA	&H00000139	
SPA_ADDTOSELECTIONEX	&H0000013A	
SPA_CLEARSELECTEX	&H0000013B	
SPA_SELECTOBJECTEX	&H0000013C	
SPA_ISPAGEVALIDFORVERSION	&H0000013D	
SPA_SELECTGRAPH	&H0000013E	
SPA_COPYABLEOBJECTSSELECTED	&H0000013f	
SPA_NONCOPYABLEOBJECTSSELECTED	&H00000140	
SPA_USABLEAREA	&H00000191	

SPA_PAGESIZEEX	&H00000193	Page size extents
SPA_EXTENTUSED	&H00000199	Extent of all objects on page, as a 4 element array
SPA_EXTENTSELECTED	&H0000019A	Extents of all selected objects, as a 4 element array
SPA_SIZESELECTEDEX	&H0000019B	Size of all selected objects, as a 2 element array

Page Object Option Constants

SPA_FLAG_SHOWUSABLE &H00000010

SPWGraphAttribute: Plot Attributes

Plot options. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SLA_BASE	&H00000300	
SLA_END	&H000003FF	
SLA_NAME	&H00000300	Determines the name of the plot
SLA_TYPE	&H00000301	Determines the type of plot. Use SLA_TYPE constants
SLA_ORGTYPE	&H00000302	
SLA_PLOTOPTIONS	&H00000303	Determines the options in effect for the plot. Use SLA_FLAG constants
SLA_NTUPLEINDEX	&H00000304	Determines the index of the Current Tuple
SLA_NTUPLEMAX	&H00000305	Returns the number of tuples present in the plot (Get only)
SLA_ADDNTUPLE	&H00000306	Adds a tuple to the plot, and selects it as the current tuple
SLA_DELNTUPLE	&H00000307	Deletes the Current Tuple
SLA_SELECTDIM	&H00000308	Determines the Current Dimension for the Plot. Use the DIM_ constants . Note that other attributes rely on the Current Dimension
SLA_HAXIS	&H00000309	Assigns or retrieves the axis used by the plot for the Current Dimension
SLA_DATACOL	&H0000030A	Determines the data column for the Current Tuple's Current Dimension
SLA_ERRORCOL	&H0000030B	Determines the error column for the Current Tuple's Current Dimension
SLA_ERRORDIRCOL	&H0000030C	Determines the error direction column for the Current Tuple's Current Dimension
SLA_ERRORDIRCALC	&H0000030D	Determines the error direction and calculation for the Current Tuple's Current Dimension
SLA_ERROROPTIONS	&H0000030E	Determines the options in effect for the Current Tuple's error bars. Use SLA_ERRF constants
SLA_HSYMBOL	&H0000030F	Gets or sets the plot's symbol object
SLA_HLINE	&H00000310	Gets or sets the plot's line object
SLA_HSOLID	&H00000311	Gets or sets the plot's solid object
SLA_SELECTFUNC	&H00000312	Determines the Selected Function; SEA attributes sent to the plot will be

		forwarded to the indicated function (as will SFA attributes)
SLA_HFUNC	&H00000313	Gets or sets the plot's Selected Function object
SLA_SAMPLERANGE	&H00000314	Indicates whether the plot should or should not sample the data point on each tuple. If off, the following sample attributes are ignored
SLA_SAMPLETOP	&H00000315	Indicates the topmost point to sample
SLA_SAMPLEBOTTOM	&H00000316	Indicates the bottommost point to sample
SLA_SAMPLEFREQ	&H00000317	Indicates the frequency of sampling. 0 = 1 = every point, 2 = every other point, etc.
SLA_LINEPATH	&H00000318	Determines the path the line takes to connect consecutive point on a line/scatter plot. Use SLA_PATH constants
SLA_REGRORDER	&H00000319	Determines the order of regression of the plot and/or all tuples on the plot
SLA_REGROPTIONS	&H0000031A	Options used to modify regression behavior. Use SLA_REGR constants
SLA_MINDATA	&H0000031C	
SLA_MAXDATA	&H0000031D	
SLA_MINDATAPLUS	&H0000031E	
SLA_MAXDATAPLUS	&H0000031F	
SLA_LINEREPEAT	&H00000320	Same as SEA_TYPEREPEAT
SLA_LINECOLUMN	&H00000321	Same as SEA_TYPECOL
SLA_SYMBOLREPEAT	&H00000322	
SLA_SYMBOLCOLUMN	&H00000323	
SLA_ERRCAPWIDTH	&H00000324	Error bar cap width
SLA_ERRTHICKNESS	&H00000325	Error bar line thickness
SLA_ERRCOLOR	&H00000326	Error bar line color
SLA_QCMETHOD	&H00000327	Reference line computation
SLA_QCTEXT	&H00000328	Reference line name/label
SLA_QCOPTIONS	&H00000329	Use SLA_QCOPTS constants
SLA_NUMCOLS	&H0000032A	
SLA_SELECTDROP	&H0000032E	
SLA_WIDTHCOLUMN	&H0000032F	Bar width worksheet column
SLA_SOLIDREPEAT	&H00000330	
SLA_SOLIDCOLUMN	&H00000331	
SLA_SHOWNAME	&H00000332	
SLA_RENDERED	&H00000333	
SLA_PIEFIRSTSLICEANGLE	&H00000334	Start position of first slice
SLA_PIEEXPLODEDSLICE	&H00000335	Use SLA_PIEEXP Constants

SLA_PIEEXPLODEFROM	&H00000336	Column for exploded slices
SLA_BARALIGNMENT	&H00000337	Use the SLA_BARALIGN constants
SLA_BARTHICKNESS	&H00000338	Individual bar widths/widths within groups
SLA_BARGROUPSPACING	&H00000339	Bar group spacing
SLA_BARGROUPSPACEFROMCOL	&H0000033A	Get group spacing from column
SLA_BARGROUPSPACECOL	&H0000033B	Worksheet column for group spacing
SLA_BARUNIFORMTHICKNESS	&H0000033C	Uniform spacing on (else as wide as possible)
SLA_BOXAVERAGE	&H0000033D	Show box plot mean line
SLA_BOX595SUMMARY	&H0000033E	Display summary symbol for 5th and 95th percentiles (else all points)
SLA_BOXCAPWIDTH	&H0000033F	Box plot whisker cap width
SLA_SUBTYPE	&H00000340	Use SLA_SUBTYPE constants
SLA_HTUPLE	&H00000341	
SLA_SUBTYPEPROPERTIES	&H00000342	
SLA_ENUMREGRFUNCS	&H00000343	
SLA_ENUMQCFUNCS	&H00000344	
SLA_ENUMLINES	&H00000345	
SLA_ENUMAXES	&H00000346	
SLA_CREATEAXIS	&H00000347	
SLA_ENUMTUPLES	&H00000348	
SLA_HDROP	&H00000349	
SLA_HNAME	&H0000034a	
SLA_NTHTUPLE	&H0000034B	
SLA_NUMTUPLES	&H0000034C	
SLA_TUPLEBYHANDLE	&H0000034D	
SLA_CONTOURLABELFREQ	&H0000034E	Contour plot label frequency
SLA_SELECTTUPLE	&H0000034F	
SLA_ERRCOLORREPEAT	&H00000350	Error bar color scheme
SLA_ERRCOLORCOL	&H00000351	Color bar color column
SLA_CONTOURFILLTYPE	&H00000358	Use the SLA_CONTFILL Constants
SLA_ERRORCOL2	&H00000359	Column for 2nd error bar value for asymmetric error bars
SLA_QUANTILEMTHD	&H00000360	Set this value to 0 to use the Cleveland method, and to 1 to use the standard statistical method
SLA_AREAFILLTYPE	&H00000361	Use the SLA_AREAFILLTYPE Constants

Statistics Constants

STAT_CONF(x)

STAT_MEAN 101

STAT_STDDEV 102

STAT_STDERR

SPWAxisTickLabelAlignment: SAA_ALIGN Constants

Axis tick label alignments.

SAA_ALIGN_DONTCARE	1.	No preference is desired, PAGEW is free to use whatever alignment it desires
SAA_ALIGN_INNER	2.	Labels should be aligned inward (consistent closeness to tick)
SAA_ALIGN_OUTER	3.	Labels should be aligned outward (consistent distance from tick)
SAA_ALIGN_CENTER	4.	Labels should be centered (center should be consistent distance from tick)
SAA_ALIGN_LEFT	5.	Labels should be left-aligned. (The left is defined relative to text)
SAA_ALIGN_RIGHT	6.	Labels should be right-aligned
SAA_ALIGN_ONDECIMAL	7.	Labels should be aligned along the decimal point (or right aligned, if no decimal is present)
SAA_ALIGN_BASE	8.	Labels should be aligned along the base (for log scales)

SPWAxisBreakMarkShape: SAA_BREAK Constants

Axis break marker shapes.

SAA_BREAK_NONE	1.	No break treatment. The only evidence of a break is the discontinuity of the axis line.
SAA_BREAK_DIAG	2.	The break is represented by two parallel diagonal lines, one on each end of the axis line where the break occurs.
SAA_BREAK_PERP	3.	The break is represented by two parallel orthogonal lines, one on each end of the axis line where the break occurs.
SAA_BREAK_S	4.	The break is represented by two "S" shaped curves, one on each end of the axis line where the break occurs.

SPWAxisTickDensity: SAA_DENS Constants

These set the probability and logit axis tick intervals.

SAA_DENS_COARSE	1.
SAA_DENS_MEDIUM	2.
SAA_DENS_FINE	3.

SPWAxisDateTimeUnits: SAA_DTUNIT Constants

Date and time axis units.

SAA_DTUNIT_INVALID	0
SAA_DTUNIT_CENTURY	1.
SAA_DTUNIT_DECADE	2.
SAA_DTUNIT_YEAR	3.
SAA_DTUNIT_MONTH	4.
SAA_DTUNIT_WEEKMON	5.
SAA_DTUNIT_WEEKTUES	6.
SAA_DTUNIT_WEEKWED	7.
SAA_DTUNIT_WEEKTHUR	8.
SAA_DTUNIT_WEEKFRI	9.
SAA_DTUNIT_WEEKSAT	1.
SAA_DTUNIT_WEEKSUN	1.
SAA_DTUNIT_DAY	1.
SAA_DTUNIT_HOUR	1.
SAA_DTUNIT_MIN	1.
SAA_DTUNIT_SEC	1.
SAA_DTUNIT_MSEC	10.

SPWAxisOptions: SAA_FLAG Constants

Axis option flags.

SAA_FLAG_AUTORANGE	&H00000008	The axis should be auto-scaled, that is, it should determine which plots are using it, and query them for their minimum and maximum values; the axis range will be some calculation based on these values
SAA_FLAG_MAJORGRID	&H00000010	The major grid line should be visible
SAA_FLAG_MINORGRID	&H00000020	The minor grid line should be visible
SAA_FLAG_MAJORGRID2	&H00000040	Show secondary major grid lines
SAA_FLAG_MINORGRID2	&H00000080	Show secondary minor grid lines
SAA_FLAG_SHOW1	&H00000100	The first of two (or four) sub-axes should be visible. This is a master control
SAA_FLAG_SHOW2	&H00000200	The second of two (or four) sub-axes should be visible. This is a master control
SAA_FLAG_NOAUTOPAD	&H00000400	Sets range to manual
SAA_FLAG_POLAR	&H00000800	Use polar axes options
SAA_FLAG_SHOW3	&H00001000	Display 3rd sub-axis
SAA_FLAG_SHOW4	&H00002000	Display 4th sub-axis
SAA_FLAG_3D	&H00040000	3D axes
SAA_FLAG_ADVANCEDRANGEOPTS	&H00100000	Use advanced range control options (separate min, max and padding controls)
SAA_FLAG_AUTOTICKS	&H00200000	Use automatic or manually determined tick intervals
SAA_FLAG_AUTORANGEMIN	&H00400000	Automatically compute axis minimum range
SAA_FLAG_AUTORANGEMAX	&H00800000	Automatically compute axis maximum range
SAA_FLAG_NOAUTOPADRANGE	&H01000000L	No automatic padding of axis range
SAA_FLAG_NOAUTOPADTICKS	&H02000000L	No padding to nearest tick mark

SPWAxisLineSelector: SAA_LINE Constants

Axis sub-line selector arguments.

SAA_LINE_ALL	0	Select all axis lines, including tick marks and breaks
SAA_LINE_AXIS	1.	The axis line itself; SEA attributes are forwarded to the axis line

SAA_LINE_MAJORTIC	2.	The major tick line object. SEA attributes are forwarded to the major tick line object. Certain SAA attributes may behave in one of two or more ways, depending on whether this or SAA_LINE_MINORTIC is selected
SAA_LINE_MINORTIC	3.	The minor tick line object. SEA attributes are forwarded to the minor tick line object. Certain SAA attributes may behave in one of two or more ways, depending on whether this or SAA_LINE_MAJORTIC is selected
SAA_LINE_MAJORGRID	4.	The major grid line object. SEA attributes are forwarded to the minor tick line object
SAA_LINE_MINORGRID	5.	The minor grid line object. SEA attributes are forwarded to the minor tick line object
SAA_LINE_BREAK	6.	The axis break object. SEA attributes are forwarded to the minor tick line object
SAA_LINE_MAJORGRID2	7.	Select major grid lines for 2nd plane (3D graphs)
SAA_LINE_MINORGRID2	8.	Select minor grid lines for 2nd plane (3D graphs)

SPWAxisMinorLogTicks: SAA_LOGTIC Constants

Common log axis minor tick options.

SAA_LOGTIC_15	&H00000001
SAA_LOGTIC_20	&H00000002
SAA_LOGTIC_25	&H00000004
SAA_LOGTIC_30	&H00000008
SAA_LOGTIC_35	&H00000010
SAA_LOGTIC_40	&H00000020
SAA_LOGTIC_45	&H00000040
SAA_LOGTIC_50	&H00000080
SAA_LOGTIC_55	&H00000100
SAA_LOGTIC_60	&H00000200
SAA_LOGTIC_65	&H00000400
SAA_LOGTIC_70	&H00000800
SAA_LOGTIC_75	&H00001000
SAA_LOGTIC_80	&H00002000
SAA_LOGTIC_85	&H00004000
SAA_LOGTIC_90	&H00008000
SAA_LOGTIC_95	&H00010000

SPWAxisPolarUnits: SAA_POLARUNIT Constants

Polar axis unit options.

SAA_POLARUNIT_DEGREES	1.
SAA_POLARUNIT_RADIANS	2.

SAA_POLARUNIT_GRADS	3.

SPWSubAxisOptions: SAA_SUB Constants

Sub-axis options.

SAA_SUB_SHOW	&H00000004	Show the axis
SAA_SUB_SHOWLINE	&H00000008	The axis line itself is drawn (if not present, the ticks will appear to float in space)
SAA_SUB_MAJOR	&H00000010	The major ticks are drawn
SAA_SUB_MINOR	&H00000020	The minor ticks are drawn
SAA_SUB_MAJORLABEL	&H00000040	The major tick labels are drawn
SAA_SUB_MINORLABEL	&H00000080	The minor tick labels are drawn
SAA_SUB_MAJORIN	&H00000100	The left/bottom axis major ticks are drawn with an inward component
SAA_SUB_MINORIN	&H00000200	The left/bottom axis minor ticks are drawn with an inward component
SAA_SUB_MAJOROUT	&H00000400	The left/bottom axis major ticks are drawn with an outward component
SAA_SUB_MINOROUT	&H00000800	The left/bottom axis minor ticks are drawn with an outward component
SAA_SUB_MAJORIN2	&H00001000	The right/top axis major ticks are drawn with an inward component
SAA_SUB_MINORIN2	&H00002000	The right/top axis minor ticks are drawn with an inward component
SAA_SUB_MAJOROUT2	&H00004000	The right/top axis major ticks are drawn with an outward component
SAA_SUB_MINOROUT2	&H00008000	The right/top axis minor ticks are drawn with an outward component
SAA_SUB_SHOWNAME	&H00010000	The axis title is shown
SAA_SUB_ALIGNMINORONTIC	&H00020000	Align on minor tick instead of with major labels
SAA_SUB_POLARLABELSINCW	&H00040000	
SAA_SUB_POLARSKIPFIRSTLABEL	&H00080000	Off and it gets MIN Val
SAA_SUB_POLARLABELBKGRND	&H00100000	Adds the background color to polar plot labels
SAA_SUB_TITLEAPEX	&H00200000	Title is drawn at ternary plot apex
SAA_SUB_MAJORLABEL2	&H00400000	The major tick labels are to be rendered for the 2nd axis
SAA_SUB_MINORLABEL2	&H00800000	The minor tick labels are to be rendered for the 2nd axis
SAA_SUB_MAJORIN3	&H01000000	The 3rd axis major ticks are drawn with an inward component
SAA_SUB_MINORIN3	&H02000000	The 3rd axis minor ticks are drawn with an inward component
SAA_SUB_MAJOROUT3	&H04000000	The 3rd axis major ticks are drawn with an outward component
SAA_SUB_MINOROUT3	&H08000000	The 3rd axis minor ticks are drawn with an outward component
SAA_SUB_MAJORLABEL3	&H10000000	The major tick labels are to be rendered for the 3rd axis

SAA_SUB_MINORLABEL3	&H20000000 The minor tick labels are to be rendered for the 3rd axis

SPWAxisTickMarkSelector: SAA_TIC Constants

Tick mark group selectors.

SAA_TIC_MAJOR	1.
SAA_TIC_MINOR	2.

SPWAxisTickLabelFormats: SAA_TLBL Constants

Axis tick label numeric formats.

SAA_TLBL_EXP	1.	All numbers are represented only by their exponent only; for example, 1000 is represented by only a '3'.
SAA_TLBL_BASEEXP	2.	All numbers are represented by a base and exponent; for example, 1000 would be represented in the form 10^3
SAA_TLBL_SCINO	3.	All numbers are represented by base and exponent; for example, 1000 would be represented by 1.0e+3
SAA_TLBL_SCINOBIG	4.	Only numbers exceeding the established threshold will be represented in SAA_TLBL_SCINO
SAA_TLBL_BASEEXPBIG	5.	Only numbers exceeding the established threshold will be represented in SAA_TLBL_BASEEXP
SAA_TLBL_ENGR	6.	All numbers are represented in engineering units, i.e., the exponent is always a multiple of three. For example, 1000 would be represented by 1.0×10^3
SAA_TLBL_ENGRBIG	7.	Only numbers exceeding the established threshold will be represented in SAA_TLBL_ENGR

SPWAxisScaleTypes: SAA_TYPE Constants

Axis scale types.

See also the SigmaPlot Help topic Axis Scale Types

SAA_TYPE_LINEAR	1.	Linear scale
SAA_TYPE_COMMON	2.	Common log scale (base 10)
SAA_TYPE_LOG	3.	Natural log scale (base e)
SAA_TYPE_PROBABILITY	4.	Probablity scale
SAA_TYPE_PROBIT	5.	Probit scale
SAA_TYPE_LOGIT	6.	Logit scale
SAA_TYPE_CATEGORY	7.	Category scale
SAA_TYPE_DATETIME	8.	Date and Time scale

SBA_SEEK Constants

SBA_SEEK_REWIND Reset the current index to the first object

SBA_SEEK_EOF Set the current index to the last object in the bag

SPWSolidFillDensity : SDA_DENS Constants

This is the solid fill pattern density. You can use any pattern density desired; the constants correspond to the values used by the user interface.

SDA_DENS_BUILTIN	0	The Windows system density
SDA_DENS_COARSE	150	A coarse pattern
SDA_DENS_MEDIUM	100	A medium density pattern
SDA_DENS_FINE	50	A fine pattern

SPWSolidFillPatterns : SDA_PAT Constants

Solid object fill patterns.

SDA_PAT_HOLLOW	1.	No pattern; the filled area of the pattern remains unchanged (transparent). A zero should be passed instead of an SDA_DENS constant if this pattern is used
SDA_PAT_SOLID	2.	A solid fill. The filled area uses the SOA_COLOR. A zero should be passed instead of an SDA_DENS constant if this pattern is used
SDA_PAT_RR	3.	Diagonal slashing rising right
SDA_PAT_RL	4.	Diagonal slashing rising left
SDA_PAT_DIACROSS	5.	Diagonal crosshatching
SDA_PAT_HORZ	6.	Horizontal banding
SDA_PAT_VERT	7.	Vertical banding
SDA_PAT_HCROSS	8.	Horizontal crosshatching

SPWLineEndOptions: SEA_END Constants

These are the options for the shape of line endings.

SEA_END_NONE	No end treatment; the line merely ends. The parameter of the SEA_LINEEND should be zero
SEA_END_ARROW	The line ends in an arrow. The of parameter the SEA_LINEEND should be an angle measurement
SEA_END_ARROWRANGE	The line ends in an arrow and an orthogonal line. The parameter of the SEA_LINEEND should be an angle measurement representing the angle of the arrow
SEA_END_BULLET	The line end is a solid circle

SPWLineTypes SEA_LINE Constants

These are options for the line type.

SEA_LINE_NONE	1.	No line
SEA_LINE_SOLID	2.	A solid, uninterrupted line
SEA_LINE_LONGD	3.	A long-dashed line
SEA_LINE_MEDD	4.	A medium-dashed line
SEA_LINE_SHORTD	5.	A short-dashed line
SEA_LINE_DOTTED	6.	A true dotted (not short-dashed) line
SEA_LINE_DASHD	7.	An alternating dash-dot pattern
SEA_LINE_DASHDD	8.	An alternating dash-dot-dot pattern

SPWGraphicPageMessages: Set Attribute Constants

GPM_SETOBJECTATTRSTRING	&H00000005
GPM_SETPAGEATTR	&H00000101
GPM_SETPAGEATTRSTRING	&H00000107
GPM_SETGRAPHATTR	&H00000201
GPM_SETGRAPHATTRSTRING	&H00000208
GPM_SETPLOTATTR	&H00000301
GPM_SETPLOTATTRSTRING	&H00000309
GPM_SETAXISATTR	&H00000401
GPM_SETAXISATTRSTRING	&H00000408

SFA_FLAG Constants

Function option flags (not yet assigned in SPW32.TLB; use the numeric values)

SFA_FLAG_FX	&H0000004	The function object is a function of *x*. Note that in a 2D coordinate systems, this option is contradictory to SFA_FLAG_FY
SFA_FLAG_FY	&H0000008	The function object is a function of *y*. Note that in a 2D coordinate systems, this option is contradictory to SFA_FLAG_FY
SFA_FLAG_FROMCOL	&H0000010	The domain should be determined by SFA_COLUMN as opposed to SFA_FROM, SFA_TO, and SFA_RESOLUTION. This flag contradicts SFA_FLAG_AUTORANGE
SFA_FLAG_AUTORANGE	&H0000020	The domain should be determined by the extent of the axis (or axes) along which the domain lies. Contradicts SFA_FLAG_FROMCOL
SFA_FLAG_PREAUGMENT	&H0000100	Pre-augment y = aug(x) + f(x) (Default is to post-augmenty = f(x) + aug(x))
SFA_FLAG_NEGAUGMENT	&H0000200	The results of the augment function should be negated before being combined with the results of the function
SFA_FLAG_NEGFUNC	&H0000400	The results of the functions should be negated before being combined with the results of the augment function
SFA_FLAG_DORMANT	&H0008000	Should it be drawn, or is it to remain unseen?
SFA_FLAG_LABELA	&H0001000	
SFA_FLAG_LABELB	&H0002000	
SFA_FLAG_MAPPED	&H0004000	Coefficient calculated on transformed units

SFA_OP Constants

SFA_OP_NOP	No operation. Produces x' = x
SFA_OP_SQRT	Square root. Produces x' = sqrt(x)
SFA_OP_ABS	Absolute value. Produces x' = abs(x)
SFA_OP_LN	Natural log. Produces x' = ln(x)
SFA_OP_LOG	Common log. Produces x' = log10(x)
SFA_OP_EXP	Natural exponent. Produces x' = ex
SFA_OP_SQUARE	Square. Produces x' = x2
SFA_OP_CUBE	Cube. Produces x' = x3
SFA_OP_MULCONST	Multiply by a constant. produces x' = x * *k* where k is determined by SFA_PREOPCONST or SFA_POSTOPCONST

SPWGraphCoordSystemType: SGA_COORD Constants

SGA_COORD_MINVAL	1.	
SGA_COORD_CART2	1.	● D Cartesian
SGA_COORD_CART3	2	3. D Cartesian
SGA_COORD_PIE	3	Pie chart
SGA_COORD_CONTOUR2	4.	2D Contour
SGA_COORD_POLAR	5.	2D Polar
SGA_COORD_TERNARY	6.	Triangle
SGA_COORD_MAXVAL	6	

SPWGraphLineSelector: SGA_LINE Constants

Used to select frame lines in 3D graphs with SGA_SELECTLINE.

SGA_LINE_ALL	0	All graph lines
SGA_LINE_FIRST	1.	
SGA_LINE_PLANES	1.	All lines in plane
SGA_LINE_FRAMEORG	2.	Origin frame lines
SGA_LINE_FRAMENONORG	3.	Non-origin frame lines
SGA_LINE_FRAMENEAR	4.	Front frame lines
SGA_LINE_FRAMEFAR	5.	Rear frame lines

SPWGraphOptions SGA_FLAG Constants

SGA_FLAG_AUTOLEGENDSHOW	&H00000004	Show the automatic legend
SGA_FLAG_TITLESUNALIGNED	&H00000400	If this is set then axis titles will placed at default positions.
SGA_FLAG_3DAXESNOTINFRONT	&H00000800	If this is set then 3D axes will be sent to front (same as SGA_AXESTOFRONT)
SGA_FLAG_AUTOLEGENDBOX	&H00001000	If this and AUTOLEGENDSHOW are both turned off, the legend is deleted and will return to default when turned back on
SGA_FLAG_AUTOLEGENDON	&H00002000	
SGA_FLAG_3DLIGHTON	&H00004000	
SGA_FLAG_GRIDINFRONT	&H00008000	If this is set the axes and grid are drawn after the plot

SPWPlotAreaFillDirections: SLA_AREAFILLTYPE Constants

Area plot fill directions

SLA_AREAFILL_NONE	0	No fill color
SLA_AREAFILL_DOWN	1.	Fill direction down to axis
SLA_AREAFILL_UP	2.	Fill direction up to axis
SLA_AREAFILL_LEFT	3.	Fill direction left to axis
SLA_AREAFILL_RIGHT	4.	Fill direction right to axis

SPWPlotBarAlignment : SLA_BARALIGN Constants

Bar chart bar alignment options.

SLA_BARALIGN_POINTLEFT	1.	Left corner at point
SLA_BARALIGN_CENTER	2.	Bar center at point
SLA_BARALIGN_POINTRIGHT	3.	Right corner at point

Missing topic found during conversion.

SPWPlotErrorBarOptions : SLA_ERRF Constants

Error bar direction and computation options

SLA_ERRF_REL0	&H00000004	Error bars are relative to zero (else absolute)
SLA_ERRF_POSFROM	&H00000008	Error bar directions are either positive, or from zero, depending on REL0 setting
SLA_ERRF_NEGTO	&H00000010	Error bar directions are either negative, or to zero, depending on REL0 setting
SLA_ERRF_GEOMETRIC	&H00000020	Use geometric mean
SLA_ERRF_FROMCOL	&H00000040	Obtain values from worksheet

SPWPlotOptions : SLA_FLAG Constants

SLA_FLAG_IGNORERANGE	&H00000004	Ignore out-of-range points
SLA_FLAG_IGNORENAN	&H00000008	Ignore missing values
SLA_FLAG_LINEON	&H00000010	Data points should be connected by lines Line/Symbol plot (not just Symbol plot)
SLA_FLAG_REGRON	&H00000020	Regressions should be calculated and updated (This is a master control)
SLA_FLAG_QCON	&H00000080	QC (aka Reference Lines) should be calculated and updated
SLA_FLAG_LINEONTOP	&H00000100	Lines should be rendered on top of symbols (otherwise they are rendered 'behind')
SLA_FLAG_REGRONTOP	&H00000200	Regressions should be rendered on top of symbols
SLA_FLAG_QCONTOP	&H00000400	QC Lines should be rendered on top of symbols
SLA_FLAG_YVERSUSX	&H00001000	Reference line direction
SLA_FLAG_DROPX	&H00002000	Symbols should have a drop line parallel to the X axis (i.e., to Y axis, or YZ plane depending on coordinate system)
SLA_FLAG_DROPY	&H00004000	Symbols should have a drop line parallel to the Y axis (i.e., to the X axis, or ZX plane depending on coordinate system)
SLA_FLAG_DROPZ	&H00008000	Symbols should have a drop line parallel to the Z axis (i.e., to the XY plane)
SLA_FLAG_WIDTHPERGROUP	&H00010000	
SLA_FLAG_FX	&H00020000	
SLA_FLAG_FY	&H00040000	
SLA_FLAG_INCRONPOINT	&H00080000	
SLA_FLAG_POLARLOOP	&H00100000	
SLA_FLAG_3DSHADINGSMOOTH	&H00200000	Gradient shading

SLA_FLAG_3DRIBBON	&H00400000 (not yet supported)
SLA_FLAG_3DFILLTRANSPARENT	&H00800000 Translucent mesh
SLA_FLAG_3DPLANECOLORACROSS	&H02000000 Stretch colors across z-range for mesh

SPWPlotFunctions: SLA_FUNC Constants

Used to return a function object. Use as an argument of the Functions property.

SLA_FUNC_NONE	0	
SLA_FUNC_FIRST	1.	
SLA_FUNC_REGR	1.	The regression line itself
SLA_FUNC_CONF1	2.	The first confidence interval line
SLA_FUNC_CONF2	3.	The second confidence interval line
SLA_FUNC_PRED1	4.	The first prediction interval line
SLA_FUNC_PRED2	5.	The second prediction interval line
SLA_FUNC_QC1	6.	The first QC (aka Reference) line
SLA_FUNC_QC2	7.	The second QC (aka Reference) line
SLA_FUNC_QC3	8.	The third QC (aka Reference) line
SLA_FUNC_QC4	9.	The fourth QC (aka Reference) line
SLA_FUNC_QC5	1.	The fifth QC (aka Reference) line
SLA_FUNC_LAST	1.	

SPWPlotLineShapeOptions : SLA_PATH Constants

Plot line shape options.

SLA_PATH_SLOPE	1.	Data points should be connected by a straight, sloping line
SLA_PATH_HORZFIRST	2.	Data points should be connected by an "L" shaped line, where the horizontal component is drawn first, followed by the vertical
SLA_PATH_VERTFIRST	3.	Data points should be connected by an "L" shaped line, where the vertical component is drawn first, followed by the horizontal
SLA_PATH_HORZCENTER	4.	Data points should be connected by an three-segment line, starting out with a vertical line extending half of the distance up or down, followed by a horizontal line, and ending in a vertical line completing the vertical distance up or down
SLA_PATH_VERTCENTER	5.	Data points should be connected by a three-segment line like HORZCENTER, except that the horizontal and vertical components are transposed
SLA_PATH_SPLINE	6.	Spline curved lines

SPWPlotPieOptions : SLA_PIEEXP Constants

Pie chart exploding slices options.

SLA_PIEEXP_NONE	1.	No exploding slices
SLA_PIEEXP_SINGLE	2.	One exploded slice
SLA_PIEEXP_COLUMN	3.	Exploded slices from worksheet column

SPWPlotRegressionOptions: SLA_REGR Constants

Plot regression line options, used as arguments for SLA_REGROPTIONS.

SLA_REGR_FORTUPLES	&H00000004	Draw regression for each curve
SLA_REGR_FORPLOT	&H00000008	Draw regression for entire plot
SLA_REGR_LINE	&H00000010	The single regression line is rendered
SLA_REGR_CONF	&H00000020	The two confidence interval lines are rendered
SLA_REGR_PRED	&H00000040	The two prediction interval lines are rendered
SLA_REGR_THRUORIGIN	&H00000080	The regression is calculated to go through the origin
SLA_REGR_99PCT	&H00000100	Confidence and prediction are 99% confidence; else 95%
SLA_REGR_TOAXES	&H00001000	All lines should be extended to the appropriate axes
SLA_REGR_INCLUDERANGE	&H00002000	All visible lines should be used to determine auto-scaling axis range (otherwise, only the data points themselves are used)
SLA_REGR_FX	&H00004000	The regressions are functions of x (horizontal)
SLA_REGR_FY	&H00008000	The regressions are functions of y (vertical)
SLA_REGR_MAPPED	&H00010000	

SPWPlotReferenceLineOptions : SLA_QCOPTS Constants

Reference line options.

SLA_QCOPTS_FX	&H00000004	X direction
SLA_QCOPTS_FY	&H00000008	Y direction
SLA_QCOPTS_LABELA	&H00000010	Show left/bottom label
SLA_QCOPTS_LABELB	&H00000020	Show right/top label
SLA_QCOPTS_INCLUDERANGE	&H00000040	
SLA_QCOPTS_SHOWQC1	&H00000100	Display first line
SLA_QCOPTS_SHOWQC2	&H00000200	Display second line
SLA_QCOPTS_SHOWQC3	&H00000400	Display third line
SLA_QCOPTS_SHOWQC4	&H00000800	Display fourth line
SLA_QCOPTS_SHOWQC5	&H00001000	Display fifth line
SLA_QCOPTS_MAPPED	&H00002000	
SLA_QCOPTS_CONSTMAPPED	&H00004000	
SLA_QCOPTS_COMPUTEMAPPED	&H00008000	

SPWPlotType: SLA_TYPE Constants

These correspond to the *base* SigmaPlot plot types. Note that these do not correspond to the Graph Wizard types; e.g., the SLA_TYPE_SCATTER type actually subsumes all scatter and line plot types found in the wizard.

SLA_TYPE_MINVAL	1.
SLA_TYPE_SCATTER	1.
SLA_TYPE_BAR	2.
SLA_TYPE_STACKED	3.
SLA_TYPE_TUKEY	4.
SLA_TYPE_3DSCATTER	5.
SLA_TYPE_MESH	6.
SLA_TYPE_PIE	7.
SLA_TYPE_CONTOUR	8.
SLA_TYPE_POLAR	9.
SLA_TYPE_POLARXY	1.
SLA_TYPE_3DBAR	1.
SLA_TYPE_TERNARYSCATTER	1.
SLA_TYPE_MAXVAL	12

SPWPlotSubtype : SLA_SUBTYPE Constants

Plot type data styles.

SLA_SUBTYPE_MINVAL	1.	
SLA_SUBTYPE_NORMAL	1.	X and Y columns can be different for each curve
SLA_SUBTYPE_VERTY	2.	All X columns must be the same
SLA_SUBTYPE_HORZX	3.	All Y columns must be the same
SLA_SUBTYPE_SUMMARYX	4.	X columns are summarized—all Y columns must be the same
SLA_SUBTYPE_SUMMARYY	5.	Y columns are summarized—all X columns must be the same
SLA_SUBTYPE_SUMMARYXY	6.	X and Y columns are summarized
SLA_SUBTYPE_FREQUENCYX	7.	Each X column is plotted against one Y value. Each Y column must be the same
SLA_SUBTYPE_FREQUENCYY	8.	Each X column is plotted against one X value. Each X column must be the same
SLA_SUBTYPE_CONSTANTX	9.	Each column is a is a row of Y values in a 3D bar chart
SLA_SUBTYPE_CONSTANTY	1.	Each column is a is a row of X values in a 3D bar chart
SLA_SUBTYPE_MAXVAL	10	

SPWSmoothingMethods: Smoothing Method Constants

SM_NEGATIVE_EXP	0	Local smoothing technique using polynomial regression and weights computed from the Gaussian density function
SM_LOESS	1.	Local smoothing technique with tricube weighting and polynomial regression
SM_RUNNING_AVERAGE	2.	Local smoothing technique that averages the values at neighboring points
SM_RUNNING_MEDIAN	3.	Local smoothing technique that computes the median of the values at neighboring points
SM_BISQUARE	4.	Local smoothing technique with bisquare weighting and polynomial regression
SM_INVERSE_SQUARE	5.	The weighted average of the values at neighboring points is computed using the Cauchy density function
SM_INVERSE_DISTANCE	6.	The weighted average of the values at neighboring points is computed using inverse distance
SM_UNWEIGHTED_REGRESSION	7.	Unweighted linear polynomial regression. Note that this option is not exposed in the user interface

SNA_REP Constants

Tuple representation options. Not yet available from SPW32.TLB; use the numeric values.

SNA_REP_UNUSED	1.	Not used; inactive, or unselected
SNA_REP_ORDINAL	2.	Use the ordinal value ('row number') of the point
SNA_REP_SCALAR	3.	One point derived from column
SNA_REP_SUMMARY	4.	Two or more points derived from column
SNA_REP_COLUMN	5.	Direct reference to column
SNA_REP_SYMBOL	6.	(For .symbol only)

SPWTupleDataSummarizations : SNA_SUM Constants

Data summarization types for error bars and box plots.

SNA_SUM_NONE	0	No error bars
SNA_SUM_MEAN	101	Mean for datapoint
SNA_SUM_MEANSTDDEV	102	Standard deviation
SNA_SUM_MEANSTDERR	103	Standard error
SNA_SUM_MEANCONF99	99	99% confidence
SNA_SUM_MEANCONF95	95	95% confidence
SNA_SUM_PERCENTILE10	210	1. th percentile
SNA_SUM_PERCENTILE25	225	25th percentile
SNA_SUM_MEDIAN	250	Median for datapoint
SNA_SUM_PERCENTILE75	275	75th percentile
SNA_SUM_PERCENTILE90	290	90th percentile
SNA_SUM_2STDDEV	300	2 standard deviations
SNA_SUM_3STDDEV	301	3 standard deviations
SNA_SUM_2STDERR	302	2 standard errors
SNA_SUM_3STDERR	303	3 standard errors
SNA_SUM_PERCENTILE	1000	Percentiles (box plot)
SNA_SUM_PERCENTILE_EX	1001	

SPWSolidShape: SOA_EXT Constants

SOA_EXT_RECT	1.	Rectanglular extent
SOA_EXT_ELLIPSE	2.	Elliptical extent

SPWAttributeRepeatType: SOA_REPEAT Constants

Settings for for attributes repeat.

SOA_REPEAT_NONE	1.
SOA_REPEAT_SAME	2.
SOA_REPEAT_AUTOINCR	3.
SOA_REPEAT_COLUMN	4.

SPWSymbolOptions SSA_Flag Constants

SSA_FLAG_VTOP	&H0000010	(Not currently supported) The symbol should be placed above data point (i.e. the Symbol's 'tack point' is at the bottom of the symbol)
SSA_FLAG_VBOT	&H0000020	(Not currently supported) The symbol should be placed below the data point (i.e. the Symbol's 'tack point' is at the top of the symbol)
SSA_FLAG_HLEFT	&H0000040	(Not currently supported) The symbol should be placed to the left of the data point (i.e. the Symbol's 'tack point' is at the right of the symbol)
SSA_FLAG_HRIGHT	&H0000080	(Not currently supported) The symbol should be place to the right of the point (i.e. the Symbol's 'tack point' is at the left of the symbol)
SSA_FLAG_FILLED	&H0000200	The symbol is filled using the color indicated by SOA_COLOR
SSA_FLAG_DOTTED	&H0000400	The datapoint is drawn as a point
SSA_FLAG_XHAIR	&H0000800	The data point is drawn as a crosshair

SPWSymbolShapes: SSA_SHAPE Constants

Symbol shapes. Use ASCII code for letters. SSA_SHAPE_TEXT for string.

SSA_SHAPE_NULL	1.	No shape. Use this in combination with SOA_OPTIONS to produce only dotted or crosshair symbols
SSA_SHAPE_CIRCLE	2.	A circle
SSA_SHAPE_SQUARE	3.	A square
SSA_SHAPE_TRIUP	4.	An upward triangle
SSA_SHAPE_TRIDN	5.	A downward triangle
SSA_SHAPE_DIA	6.	A diamond
SSA_SHAPE_HEX	7.	A hexagon
SSA_SHAPE_HBAR	8.	Horizontal bar
SSA_SHAPE_VBAR	9.	Vertical bar
SSA_SHAPE_TEXT	&H0000FFFF	Specified string

SPWTextFlags: STA_FLAG Constants

Text object options for STA_OPTIONS.

STA_FLAG_BOLD	&H00000004	The default font is to be bold; STA_BOLD is preferred
STA_FLAG_ITALIC	&H00000008	The default font is to be italicized; STA_ITALIC is preferred
STA_FLAG_UNDERLINE	&H00000010	The default font is to be underlined; STA_UNDERLINE is preferred
STA_FLAG_RELATIVE	&H00004000	The text is to be rotated relative to the reference angle set by STA_RELANGLE
STA_FLAG_RELTACK	&H00002000	The text is to be placed relative to its tack point
STA_FLAG_VISIBLE	&H00008000	Hides the label
STA_FLAG_LGNDRIGHT	&H00000100	True to position a legend to right, rather than left of text
STA_FLAG_BKOPAQUE	&H00000200	Opaque background for text

STA_SELECT Constants

Not supported by SPW32.TLB. Use the numeric values instead.

STA_SELECTEND	&H000FFFF	Used in place of the MAKELONG parameter to select the entire string.
STA_SELECTOFF	&H000FFFF	Used in place of the MAKELONG parameter to deselect the entire string.
STA_SELECTALL	&HFFFF0000 (a MAKELONG (0,STA_SELECT_LAST))	Used in place of the end index in the MAKELONG macro to represent the last character in the string

STA_LEGENDSTYLE Constants

Legend style options. Not supported by SPW32.TLB; use the numeric values instead.

STA_LEGENDSTYLE_LINE	1.	Only the line used in the curve is shown.
STA_LEGENDSTYLE_SYMBOL	2.	Only the symbol used in the curve is shown.
STA_LEGENDSTYLE_BOTH__O_	3.	Both symbol and line are shown in the legend with the symbol in the center of the legend.
STA_LEGENDSTYLE_BOTH_O_O	4.	Both symbol and line are shown in the legend with two symbols at either end of the legend.
STA_LEGENDSTYLE_REGR	5.	
STA_LEGENDSTYLE_QC	6.	

SPWGraphAttribute: Solid Attributes

Solid options. Solids include graph planes, bars, and drawn solids objects. These are typically the values of the first or second arguments (respectively) set using the SetCurrentObjectAttribute or SetAttribute methods.

SDA_BASE	&H00000800	
SDA_END	&H000008FF	
SDA_PATTERN	&H00000801	Determines the pattern to be used. Use SDA_PAT and SDA_DENS constants
SDA_EDGELINE	&H00000802	Assigns the line attribute to be used for the outline of the solid area
SDA_COLOR	&H00000803	Sets the STOCKSCHEME pattern used for the solid
SDA_ALTCOLOR	&H00000804	
SDA_EDGECOLOR	&H00000805	Assigns the color to be used for the outline of the solid area
SDA_OPTIONS	&H00000806	
SDA_PATTERNREPEAT	&H00000807	Sets the SPWStockScheme pattern used for the solid
SDA_COLORREPEAT	&H00000808	Sets the STOCKSCHEME fill color used for the solid
SDA_EDGECOLORREPEAT	&H00000809	Sets the STOCKSCHEME edge/pattern color used for the solid
SDA_PATTERNCOL	&H0000080A	Sets the source column used for fill pattern
SDA_COLORCOL	&H0000080B	Sets the source column used for fill color
SDA_EDGECOLORCOL	&H0000080C	Sets the source column used for edge/pattern color
SDA_EDGETHICKNESS	&H0000080E	Sets the edge line thickness. Same as SEA_THICKNESS

SDA_DENSITYREPEAT	&H0000080F	Sets the pattern density scheme
SDA_DENSITYCOL	&H00000810	Sets the pattern density column
SDA_FLAG_HIDE	&H00000004	Sets the solid to be hidden

SPWGraphAttribute: Symbol Attributes

SSA_BASE	&H00000700	
SSA_END	&H000007FF	
SSA_SIZE	&H00000701	Determines the size of the symbol
SSA_SHAPE	&H00000702	The shape of the symbol. Use SSA_SHAPE constants
SSA_SIZEREPEAT	&H00000703	Sets STOCKSCHEME for symbol size. Not yet supported
SSA_SHAPEREPEAT	&H00000704	Sets STOCKSCHEME for symbol shape
SSA_SIZECOL	&H00000705	Worksheet column for symbol size
SSA_SHAPECOL	&H00000706	Worksheet column for symbol shape
SSA_OPTIONS	&H00000707	Modifies behavior of symbols. Use SSA_FLAG constants
SSA_EDGECOLOR	&H00000708	The color of the edge of the symbol
SSA_EDGETHICKNESS	&H00000709	The thickness of the symbol edge
SSA_COLOR	&H0000070A	The symbol fill color (identical to SOA_COLOR)
SSA_STRING	&H0000070B	The symbol string
SSA_FONT	&H0000070C	The symbol font
SSA_COLORREPEAT	&H0000070D	Sets STOCKSCHEME for symbol fill color
SSA_EDGECOLORREPEAT	&H0000070E	Sets STOCKSCHEME for symbol edge color
SSA_COLORCOL	&H0000070F	Worksheet column for symbol fill color
SSA_EDGECOLORCOL	&H00000710	Worksheet column for symbol edge color
SSA_ORIENTATION	&H00000712	
SSA_HTEXT	&H00000713	Text symbol

SPWGraphAttribute: Text Attributes

STA_BASE	&H00000500	
STA_END	&H000005FF	
STA_TEXT	&H00000500	Get/Set the text to be contained within a label. This is plain text only, and cannot represent any textual styles (i.e. font, color, bold, etc.). See STA_RTF. If the result is NULL, then the length of the string is returned
STA_LENGTH	&H00000501	The length of the text; expressed as a byte count. [This is a Get-Only attribute.] This is only the number of printable characters. See STA_RTF for determining the

		storage requirements
STA_INITFONT	&H00000502	The initial font to use for the label; the label text may encode font change information itself, this merely provides a default
STA_INITSIZE	&H00000503	The initial font size to use for the label; the label text may encode font size change information; this provides a default
STA_ORIENTATION	&H00000504	Set/Get the absolute rotation of the text label, in tenths of degrees. Note that this works regardless of the setting of the STA_FLAG_RELATIVE
STA_OPTIONS	&H00000505	Options that modify the behavior of the text. Use STA_FLAG constants
STA_SELECT	&H00000506	A MAKELONG of the beginning and ending position of the 'selection.' Adheres in behavior to Windows edit controls; used to manipulate a portion of the text without accessing the whole string. Use the STA_SELECT constants
STA_SELECTEDTEXT	&H00000508	Gets/Sets the text represented by the selection. Only plain text without formatting information is passed
STA_RELANGLE	&H00000509	Gets/Sets the reference angle from which relative text label rotation is measured, in tenths of degrees. Note that changing this parameter only has an immediate affect if the flag, STA_FLAG_RELATIVE, has been set using the STA_OPTIONS command
STA_RELORIENTATION	&H0000050A	The rotation of the text label, in tenths of degrees, relative to the current reference angle set by STA_RELANGLE. Note that this works regardless of the setting of the STA_FLAG_RELATIVE; and is added to the reference angle and that becomes the new absolute rotation
STA_RTF	&H0000050B	Gets/Sets the label to the Rich Format Text string pointed to. In the case of the Get, if NULL, then the actual length of the string, including RTF formatting characters is returned
STA_SELECTEDRTF	&H0000050C	Gets/Sets the text representing the selection. The string is interpreted or formatted as an RTF string
STA_BOLD	&H0000050D	Set: If TRUE, then the current selection region is made **bold**, otherwise, it is made entirely non-bold. Get: Returns the state of the STA_FLAG_BOLD flag
STA_ITALIC	&H0000050E	Set: If TRUE, then the current selection region is made *italic*, otherwise, it is made non-italic. Get: Returns the state of the STA_FLAG_ITALIC flag
STA_UNDERLINE		Set: If TRUE, then the current selection region is underlined, otherwise, any underlining in the selection region is removed. Get: Returns the state of the STA_FLAG_UNDERLINE flag
STA_COLOR	&H0000050F	Set: Changes the text in the current selection region to the specified color. A maximum of eight colors per label are allowed. After that, both the selected text, and any text using the eighth color are changed to the specified color: Get: Returns the default color.
STA_FONT	&H00000510	Set: Changes the text in the current selection region. A maximum of eight fonts per label are allowed. After that, both the selected text, and any text using the eighth font are changed to the specified font. Get: Returns the value of STA_INITFONT
STA_SIZE	&H00000511	Sets the text in the current selection region to the specified size in height (in 1000ths of an inch). Any number of font sizes may occur in any given label
STA_ALIGN	&H00000512	Sets the alignment with respect to the tackpoint. For paragraph alignment, use STA_PARAGRAPHJUSTIFY. Uses the Text Justification Constants
STA_TACKPOINT	&H00000513	Gets/Sets the current absolute location of the text string. This always works, regardless of the state of the STA_FLAG_RELTACK flag
STA_RELORIGIN	&H00000514	Gets/Sets the current relative origin of a text object. The text object always remembers this value, but it will not move upon receiving this message unless the STA_FLAG_RELTACK flag has been set with the STA_OPTIONS command
STA_RELTACKPOINT	&H00000515	Gets/Sets the current location of the label, relative to the last relative origin set with the STA_RELORIGIN command. This always works, regardless of the state of the STA_FLAG_RELTACK flag
STA_TEXTSTATE	&H00000516	
STA_SCRIPT	&H00000517	
STA_HPLOTLEGEND	&H00000520	Gets/Sets the handle of the plot containing the symbol or line to be used for the legend to be displayed with the label. If NULL, then no legend will displayed
STA_LEGENDTUPLENO	&H00000521	Gets/Sets the ordinal number of the curve (tuple) within the plot (specified with STA_HPLOTLEGEND) which contains the symbol for which a legend is desired
STA_LEGENDSYMBOLNO	&H00000522	Gets/Sets the ordinal number of the symbol within the tuple (STA_LEGENDTUPLENO) within the plot (STA_HPLOTLEGEND) which is to be used

		as a legend for the text label
STA_LEGENDSTYLE	&H00000523	Specifies the style of the legend. Use one of the STA_LEGENDSTYLE constants
STA_LEGENDGRAPHNO	&H00000524	
STA_LEGENDPLOTNO	&H00000525	
STA_UNDERLINE	&H00000526	
STA_AUTOTEXT	&H00000527	
STA_LEGENDINFO	&H00000528	
STA_LEGENDTEXT	&H00000529	
STA_LEGENDBITMAP	&H0000052A	
STA_LINESPACING	&H0000052B	Sets the line spacing between paragraphs
STA_PARAGRAPHJUSTIFY	&H0000052C	Sets the alignment of a paragraph. Uses the Text Justification Constants
STA_LEGENDISLINESYMBOL	&H0000052D	
STA_LEGENDLINETYPE	&H0000052E	
STA_LEGENDLINECOLOR	&H0000052F	
STA_LEGENDSYMBOLSHAPE	&H00000530	
STA_LEGENDSYMBOLFLAGS	&H00000531	
STA_LEGENDSYMBOLEDGECOLOR	&H00000532	
STA_LEGENDSYMBOLFILLCOLOR	&H00000533	
STA_LEGENDSOLIDPATTERN	&H00000534	
STA_LEGENDSOLIDFILLCOLOR	&H00000535	
STA_LEGENDSOLIDEDGECOLOR	&H00000536	

SPWTextJustifications Text Justification Constants

Note that these values are used to set both the alignment to the tackpoint (STA_ALIGN) and the paragraph alignment (STA_PARAGRAPHJUSTIFY).

STA_JUSTIFY_CENTER 1

STA_JUSTIFY_LEFT 2

STA_JUSTIFY_RIGHT 3

SPWGraphAttribute: Tuple Attributes

Plotted columns (tuple) options.

SNA_BASE	&H00000900

SNA_END	&H000009FF	
SNA_OPTIONS	&H00000902	
SNA_SELECTDIM	&H00000903	Determines whether following applies to x, y, or z
SNA_ORGTYPE	&H00000905	
SNA_REWIND	&H00000906	
SNA_DATACOL	&H00000908	Column data comes from
SNA_ERRORCOL	&H00000909	Column error bar data comes from
SNA_ERRORDIRCOL	&H0000090A	Column error bar direction comes from
SNA_NUMCOLS	&H0000090B	
SNA_MINDATA	&H0000090C	
SNA_MAXDATA	&H0000090D	
SNA_MINDATAPLUS	&H0000090E	
SNA_MAXDATAPLUS	&H0000090F	
SNA_SIZE	&H00000910	
SNA_ROWSTEP	&H00000911	
SNA_COLSTEP	&H00000912	
SNA_FIRSTROW	&H00000913	
SNA_LASTROW	&H00000914	
SNA_REPTYPE	&H00000915	Type of representation; use the SNA_REP constants
SNA_ORDINALNUMBER	&H00000916	
SNA_SUMMARYMETHOD	&H00000917	Tuple data summarizations . Use SNA_SUM constants

Worksheet Border Constants

Set the thickness of selected worksheet cell borders.

SPW_BORDER_DEFAULT	&H00000000
SPW_LEFT_THIN	&H00000001
SPW_TOP_THIN	&H00000002
SPW_RIGHT_THIN	&H00000004
SPW_BOTTOM_THIN	&H00000008
SPW_LEFT_MEDIUM	&H00000010
SPW_TOP_MEDIUM	&H00000020
SPW_RIGHT_MEDIUM	&H00000040

SPW_BOTTOM_MEDIUM	&H00000080
SPW_LEFT_THICK	&H00000100
SPW_TOP_THICK	&H00000200
SPW_RIGHT_THICK	&H00000400
SPW_BOTTOM_THICK	&H00000800

Macro Examples

Area Below Curves

Border Plots

Insert Graphs into Word

Label Symbols

Merge Columns

Paste to PowerPoint Slide

Quick Re-Plot

Rank and Percentile

Survival Curve

Batch Process Excel Files

Color Transition Values

Compute 1st Derivative

Frequency Plot

Gaussian Cumulative Distribution

Piper Plots

Plotting Polar and Parametric Equations

Power Spectral Density

Vector Plot