

Super 2d/3d Graph Library

.NET Edition

Version 12.5

Charting component for .NET applications

© 2006-2009 Software Siglo XXI
www.softwaresigloXXI.com

REFERENCE GUIDE

TABLE OF CONTENTS

Introduction.....	7
Version history.....	8
Installation and Deployment.....	14
Library Usage	15
Windows application projects.....	15
ASP.NET application project (web application)	16
Populating the Chart and the SeriesFactory helper class	19
Tips.....	20
Advanced feature: combo charts	21
Advanced feature: SVG charts (vector charts)	23
Advanced feature: PDF export (... and chart printing)	27
Advanced feature: serialization	28
Sample code for Windows application (VB.NET)	29
Sample code for C# (Windows application)	31
Sample code: creating charts without using forms (C#)	32
Library Quick Reference	33
Properties.....	33
Methods	39
Events	39
Library Reference	40
Animate	40
Autorefresh	41
BackColor	42
BackStyle.....	43
BorderColor	44
BubbleMaxRadius	45
BuiltinDialogAllowed	46
BuiltinDialogLanguage	47
Calculation	48
CalculationLineStyle	49
CastShadows	50
ChartAddNew	51
ChartAdd	52
ChartItem.....	53
ChartRemove.....	54

CreateAnimatedGIF	55
CreatePDF	56
CreateSVG	57
DataSource	58
DataSourceMode	59
DataTableBackColor	60
DataTableColor	61
DataTableFont	62
DataTableText	63
Depth	64
GanttSetToday	65
GanttSetTodayLabel	66
GaugeBackStyle	67
GaugeBigTickStyle	68
GaugeBorderStyle	69
GaugeCentralCircleBackStyle	70
GaugeCentralCircleBorderStyle	71
GaugeCentralCircleRadius	72
GaugeCentralCircleRadius	73
GaugeMarginFactor	74
GaugeNeedleStyle	75
GaugeSmallTickStyle	76
GridBackgroundImage	77
GridBackgroundImageTransparency	78
GridLineStyle	79
HoleSize	80
HotSpotBackColor	81
HotSpotBackColor	82
HotSpotColor	83
HotSpotsCrossLineStyle	84
HotSpotsEnabled	85
HotSpotsDateTimeFormat	86
HotSpotsNumericFormat	87
Image	88
ItemsHighlight	89
ItemsLegend	90
LegendAlignment	91

LegendBackColor	92
LegendColor	93
LegendFont	94
LegendText	95
LicenseeIdentifier	96
MarginBottom	97
MarginLeft	98
MarginRight	99
MarginTop	100
MarkHighest	101
MarkLowest	102
PieBorderStyle	103
PointSize	104
PointStyle	105
Quality	106
Reset	107
Resolution	108
Series	109
SeriesColor	110
SeriesBorderStyle	111
SeriesFillStyle	113
SeriesLegend	114
SeriesLineStyle	115
SeriesPicture	116
SettingsFromString	117
ShadowColored	118
ShadowOffset	119
ShadowSmooth	120
ShowDataTable	121
ShowDividerX	122
ShowItemsLegend	123
ShowLegend	124
ShowLegendInDataTable	125
ShowPoints	126
ShowSolidColors	127
ShowSurface3d	128
ShowValues	129

ShowValuesMax	130
ShowValuesMin	131
ShowXAxis	132
SlantX	133
SlantY	134
Style	135
Thickness	136
Title	137
TitleColor	138
TitleFont	139
ValuesAlignment	140
ValuesBackColor	141
ValuesColor	142
ValuesFont	143
ValuesFormat	144
ValuesLocationStyle	145
ValuesStyle	146
WallsBackColorPrincipal	147
WallsBackColorSecondary	148
WallsBackStyle	149
WallsStyle	150
XAxisColor	151
XAxisDividersCount	152
XAxisHeight	153
XAxisLabelsRotated	154
XAxisLabelsRotated90	155
XAxisNumericFormat	156
XAxisScaleMaximum	157
XAxisScaleMaximumDateTime	158
XAxisScaleMinimum	159
XAxisScaleMinimumDateTime	160
XAxisScaleMode	161
XAxisTitle	162
XAxisTitleFont	163
XAxisValuesFont	164
YAxisColor	165
YAxisDividersCount	166

YAxisIndicators167

YAxisLabelsRotated.....168

YAxisLocation169

YAxisNumericFormat.....170

YAxisScaleMaximum171

YAxisScaleMinimum172

YAxisScaleMode.....173

YAxisShowDividers.....174

YAxisTitle.....175

YAxisTitleFont.....176

YAxisValuesFont.....177

Introduction

Super2d3dGraphLibrary is a native .NET library for chart rendering. From version 4, can be used either as an user control (draggable on a WinForm) and as a rendering component for use in ASP.NET web applications (although it's not a web control).

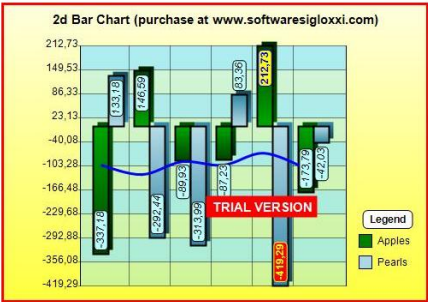
Super2d3dGraphLibrary allows you to render different nice charts with great flexibility. Using its properties and methods you can show or hide any element, or display them with the look you prefer. Current version allows several types of charts, like bars, stacked bars, scatter, bubble, spline, area and pie. The library also supports several 3D chart types like bars, stacked area, line and pie. Please refer to www.softwaresigloxxi.com for a list with the current charts available.

This document explains in a summarized form the functionality of the library, its properties and methods.

If you need further help about anything, don't hesitate to contact us at contact@softwaresigloxxi.com or on our website at www.softwaresigloXXI.com.

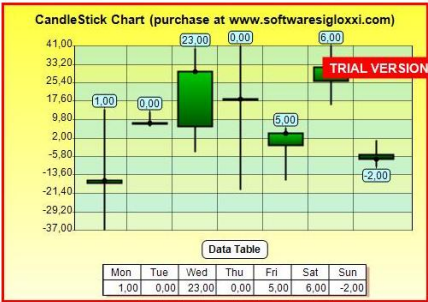
We heartly recommend to take a look at our tutorials and sample code at our website since this document becomes quickly obsolete.

Sample 1



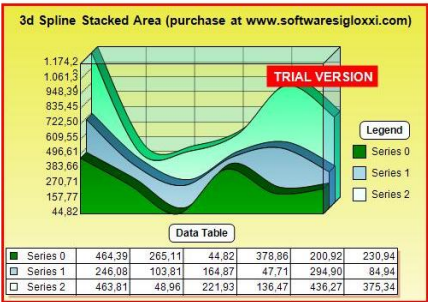
Press F5 to draw another random chart!

Sample 2

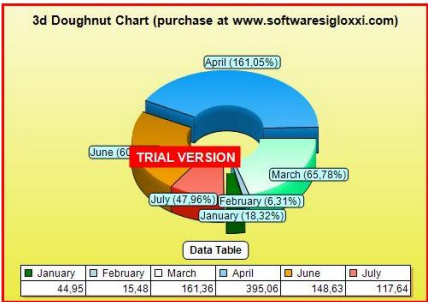


Press F5 to draw another random chart!

Sample 3



Sample 4



Version history

Version 12.5

- Added Databinding support
- Ability to select chart items and move them with mouse
- Ability to rotate XYZ charts using mouse
- New submenu in context menu "Selection" with several options
- Added HotSpotOutlineEnabled and HotSpotOutlineStyle
- Added DataTableAdditionalRowTitle and DataTableAdditionalValues
- Added ShowFrontLayer and FrontLayerColor
- Added SurfaceWireOnly
- Some bugs fixed

Version 12.1

- New chart style: Tornado chart!
- Improved multi-gauge charts
- Some bugs fixed

Version 12.0:

- Improved Gauge charts appearance with Range Colors (see new properties like GaugeRangeColors)
- New SeriesGap property to allow specify the factor of space between series in a bar chart (from 0 to 1).
- Support for Hatch styles in Bar and Pie charts
- Ability to read data from Excel files in XYZ charts
- Can specify empty values (Nothing values or use SeriesFactory's method AddValueEmpty)
- New values back style: shadowed (use ValuesBackColor to specify shadow color)
- Smart location of values in 2D Pie chart improved
- Using Calculation = AVERAGE in XY charts draws a Trend Line between 2D points.
- New CalculationTrendLine property (returns equation of trend line $y=bx+a$)
- Can hide YAxisIndicator caption
- Some bugs fixed

Version 11.5:

- SeriesFactory class: new method AddPointsFromFile (reads XYZ data from Excel file)
- Can set background image for entire control (before background image worked only for grid's background)
- New properties: SeriesPointSize, SeriesPointStyle for setting size and style of points of each serie in XY charts
- New context menu option to save chart to disk in animated GIF format
- Some context menu options now can take effect on only one chart in a combo chart
- Fixed: incorrect date/time string format in X-axis
- Improved X/Y/Z charts
- Can draw Surface chart from any number of random XYZ points using Delaunay triangulation algorithm

Super 2d/3d Graph Library Reference

Version 11:

- New chart types: 3D XYZ Scatter, 3D XYZ Lines, 3D XYZ Mesh, 3D XYZ Surface
- Ability to rotate chart in 3D (only XYZ charts)
- Improved smart setting for caption/values location
- Added lot of new options to context menu
- Added YAxisTickLength, XAxisTickLength, ZAxisTickLength, ShowYAxis, ShowZAxis, ZAxisScaleMode, ZAxisScaleMinimum, ZAxisScaleMaximum, ZAxisValuesFont, ZAxisDividersCount, ZAxisNumericFormat, ZAxisColor and other Z axis related properties
- SeriesFactory: added formula function in new AddPoints method
- SeriesFactory: added AddValueCaption method to specify a string for value captioning instead of normal numeric value (numeric value can be preserved using special tag \$\$ in the string passed). Caption can be hidden assigning "" to any ValueCaption or include the original value in a phrase using \$\$ reserved tag.
- Added ValueClick mouse event

Version 10.7:

- New chart type: 2D Stacked Spline Area
- New properties: DataTableStartIndex / DataTableLastIndex for selecting a range of values to display in the datatable (base 1). Default to zero to show all values.
- Fixed legend items order in stacked charts

Version 10.6:

- New property: PieBorderStyle for setting a custom border for pie segments (2D/3D)
- Changed implementation of YAxisIndicators
- Rendering and control handling in Visual Studio .NET is faster now!
- Fixed: memory leak caused VS IDE crash when using many controls on same Windows Form
- Fixed: tooltips don't show correctly in X/Y charts
- Fixed: some charts appear distorted when exported to PDF
- Some minor bugs fixed

Version 10.5:

- New chart type: X/Y line
- Improved pie and doughnut labels location for SMART setting
- Can specify min and max range for values' caption visibility (ShowValuesMax and ShowValuesMin properties)

Version 10.4:

- New chart type: Radar chart
- GridLineStyle property added

Version 10.3:

- Complete serialization/deserialization functionality (see SettingsFactory class)
- Reset method added
- Improved tooltips (HotSpots)
- Improved context menu
- Some bugs fixed

Super 2d/3d Graph Library Reference

Version 10.2:

- Ability to export/import look and feel settings (SettingsFromString and Settings class)
- Chart Wizard Tool included in package (helps customizing charts and generates code)
- Some bugs fixed

Version 10.1:

- New chart type: Gantt chart
- Improved smart positioning algorithm of values
- Some bugs fixed

Version 10.0:

- New feature! Export any chart to PDF format
- New property: SeriesBorderStyle
- Clone method implemented (returns an independent copy of the chart control)
- Tooltips enhanced (see new Tooltip* properties)
- Some bugs fixed

Version 9.5:

- New chart styles: STYLE2D_GAUGE, STYLE2D_XY_SPLINE
- New feature! Y-AXIS INDICATORS. Show referente lines over the chart.

Version 9.2:

- New chart style: STYLE2D_LINE_DATETIME

Version 9.1:

- New chart style: STYLE2D_SCATTER_DATETIME
- New additional calculation style for Scatter and Bubble: CALCULATION_AVERAGE
- New scale mode for X/Y axis: SCALEMODE_AUTOMATIC_MINIMUM_ZERO
- Improved values location algorithm for 2D pie chart
- Minor bugs fixed

Version 9.0:

- New major feature: support for SVG (Scalable Vector Graphics) format
- New major feature: hotspots (tooltips)
- New X-axis scale mode: DateTime
- Change datatable and legend font: see DataTableFont, LegendFont properties

Version 8.6:

- New point style: triangle
- Fixed initialization of some arrays
- Fixed fill style for 2d stacked area
- Fixed legend location when set to top/middle or bottom/middle

Version 8.5:

Super 2d/3d Graph Library Reference

- New major feature: create high quality animated gifs!
- New chart types: 2d fixed stacked bars, 3d stacked bars, 3d stacked columns.
- New animation style: STYLEANIMATION_WAVE
- Enhanced shadows (see ShadowColored, ShadowSmooth and ShadowOffset properties)
- Can hide Y-Axis dividers (see YAxisShowDividers property)
- X-Axis labels can now be rotated either 90° or 45°
- Can make chart control transparent
- Many small refinements

Version 8.0:

- New major feature: combo charts!
- New chart type: 2d horizontal bar
- May set the height of the space allocated for X-Axis labels
- May specify the location of the Y-Axis: left or right
- Set individual colors for Y-Axis and X-Axis values (YAxisColor and XAxisColor properties)
- New LegendBackColor property
- Can specify the number of elements to be shown in X-Axis (default value is 0 = all)
- Some bug fixes

Version 7.5:

- Can specify individual fonts for each axis (YAxisValuesFont and XAxisValuesFont)
- May set the inner margins for any lateral (MarginBottom, MarginLeft, MarginRight and MarginTop)
- SeriesFactory helper class
- Lot of improvements and some bugfixes
- Extended Reference Guide

Version 7.1:

- A few bug fixes and improvements

Version 7.0:

- New chart types: 2d and 3d cylinder chart, 3d spline chart, 2d rounded bars, 3d spline area and 3d spline stacked area!
- Enhanced 3d stacked area
- New additional calculations: maximum line and spline maximum line
- Corrections for 3d doughnut chart
- Built-in contextual dialog: allow your users to customiza the chart in real-time with no code!! (only winforms)
- English and Spanish languages supported
- Minor enhancements and bugfixes

Version 6.5:

- New chart type: 2d and 3d doughnut charts (with explode option)
- Specify both axis titles and fonts (YAxisTitle, YAxisTitleFont, XAxisTitle and XAxisTitleFont)
- Specify depth (thickness) for pie or doughnut charts (Depth property)
- Specify position of legend (LegendAlignment property)
- Specify hole size for doughnut charts (HoleSize property)

Super 2d/3d Graph Library Reference

- Show or hide additional points or marks over line, spline and area charts (ShowPoints property)
- Minor enhancements to some charts and bugfixes

Version 6.0:

- New chart types: pyramid and picture chart!
- New background gradient styles.
- Can specify numeric format for both axis.
- Choose between automatic or fixed scale mode.
- Grid supports background pictures with transparency.
- Enhanced 2D and 3D exploded pie charts.

Version 5.5:

- Better text rendering options (if quality property is set to HighQuality, ClearType is activated for all text rendering; "antialias" is also supported now)
- X and Y axis labels can be rotated 45° (XAxisLabelsRotated and YAxisLabelsRotated)
- Option to mark highest and lowest values (MarkHighest and MarkLowest)
- Minor enhancements to some charts and bugfixes

Version 5.0:

- Two new 2d charts: scatter and bubble.
- Possibility to specify fill-style for grid: transparent, bands or chess-pattern.
- Aesthetic enhancements to data table with possibility to specify colors for text and background.
- Inside the bars values are drawn bottom-up direction and in cursive for better readability.
- New properties for points (PointSize and PointStyle).
- Better line drawing
- Smart values location: avoid label overlap (new value for ValuesLocationStyle)

Version 4.0:

- Chart library can be used in ASP.NET web applications.
- Sample ASP.NET application.
- Some minor enhancements and bug-fixes.

Version 3.0:

- Can assign colour and gradient style to chart background.
- Can show an additional and automatic calculation over the main chart (an average spline or PARETO chart).
- Show or hide values on X axis.
- New 2d chart styles: high-low, high-low-close, candlestick y spline.
- Can specify values location: Standard or inside the bar.
- New events.

Version 2.0:

- Renders the chart over the form fast and reflects changes in properties in real time.
- Can specify output bitmap resolution (great for printing; by default, screen resolution is used).

Super 2d/3d Graph Library Reference

- Allows exporting the bitmap in any of the following formats: BMP, GIF, JPG, PNG, TIFF y WMF.
- Can show/hide any element (data table, legend, horizontal dividers, vertical dividers, values, ...)
- Assign any style, font, color or gradient to the chart.
- Many chart styles and more coming in each version of the library
- Automatic or manual refresh when changing a property of the user control.
- Animation: one call to Animate() allows to animate the chart, running each serie from 0 to the original values.
- Can specify X or Y slant for 3d charts.
- Optional can show surface line on 3d charts (a nice graphical division from positive and negative values).
- Can show/hide walls on 3d charts.

Installation and Deployment

The setup application will install the library in the Global Assembly Cache so it will be available to both web and desktop applications. If you want to manually deploy the library to a production server you should copy the single DLL to any folder and register the library into the Gac with the following command:

```
Gacutil -i Super2d3dGraphLibrary.dll
```

The same library may be used in both web and desktop applications. Each type of application have a different usage of the library.

The library has no other dependency than .NET Framework so to deploy the component just copy the single DLL file to the destination folder (register into the GAC if necessary).

Important note:

If you have previously installed trial version and when using licensed version "Trial Version" continues showing on the chart, please do the following:

- 1.- Uninstall trial version from Control Panel – Add/remove programs.
- 2.- Copy the licensed version to desired folder and register it into the GAC with gacutil if necessary.
- 3.- Remove current reference to trial version and point to the licensed DLL.

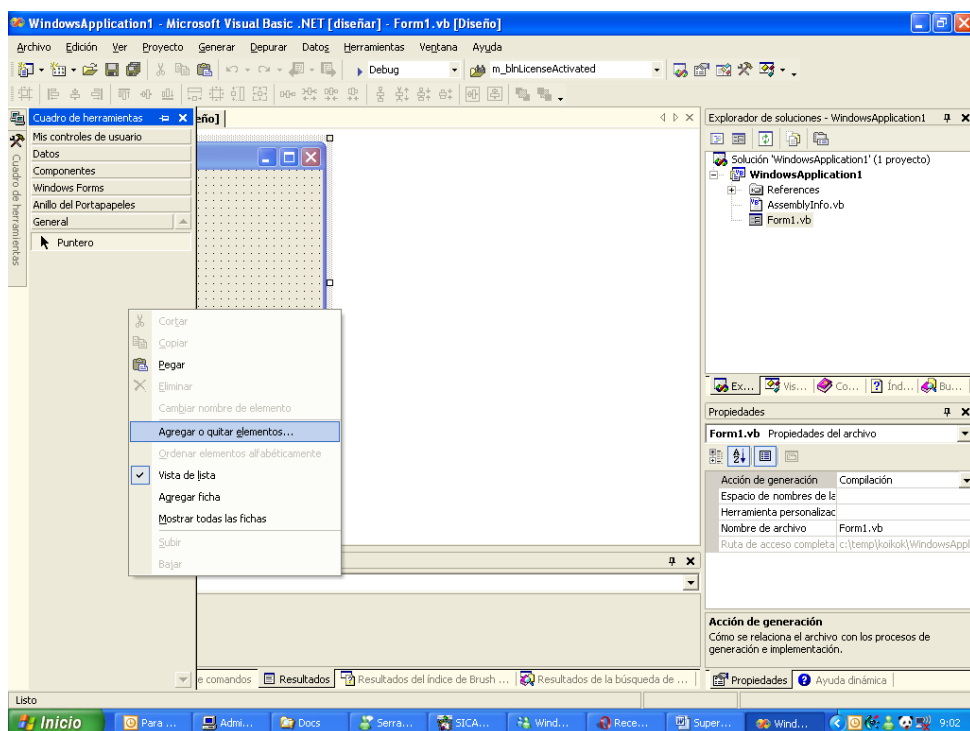
Library Usage

In desktop applications, you add the control to the toolbox window of the Visual Studio IDE. In web applications all the chart functionality must be used directly with code since the library is not a web control. However, this way you can use the library both in ASP.NET and ASP applications provided you have .NET Framework installed.

Let's begin with a Windows application project:

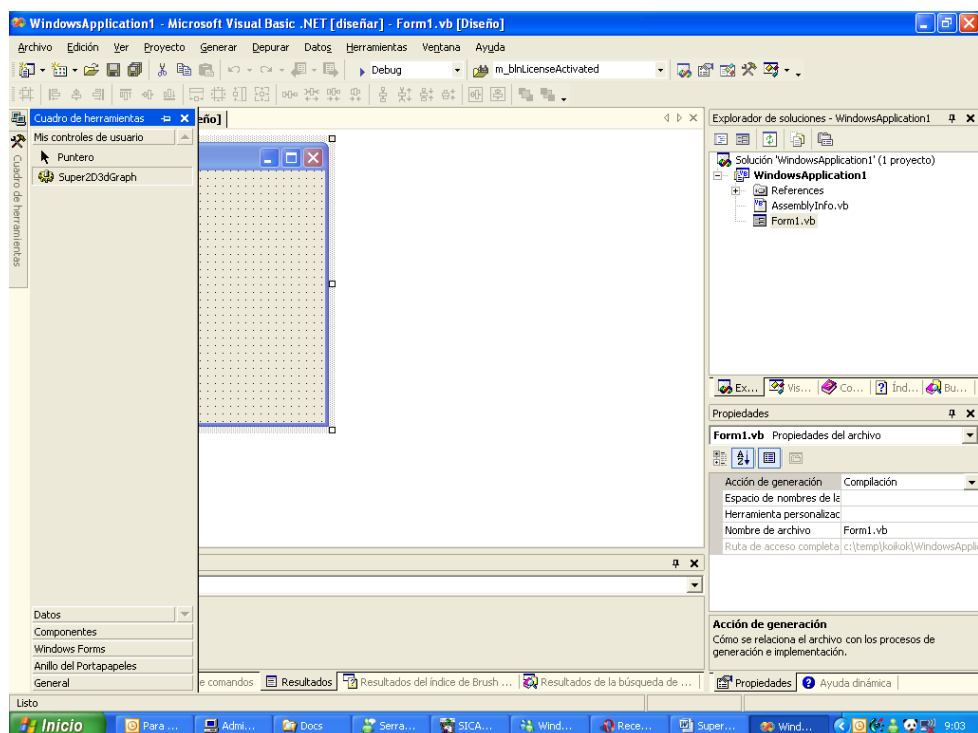
Windows application projects

To use this library as an user control in a WinForm, you only need to add it to the toolbox palette. Click right button over the toolbox palette and select Add/Remove elements...:



Next, if we have registered the component with gacutil, it should be listed on the available component list. If not, click on Browse... button to locate Super2d3dGraphLibrary.dll file and then you will be able to see it on the toolbox, ready to be dragged over the form:

Super 2d/3d Graph Library Reference



ASP.NET application project (web application)

In ASP.NET projects, the operation of the component is different, because it's not a web control. Only two steps are needed to display charts on your web pages. First, you need to create a asp.net page that instantiate the library and renders the chart to the browser. Second, you need to reference this asp.net page in the web page you want the chart to appear.

For example, assume that you want to show a chart on the page *webform1.aspx*. In this page, you only need to include an HTML element like this:

```

```

SampleChart.aspx will be the page that will instantiate of chart library, calls its members and renders the chart on the Page_load event. This way, chart will be shown on your page without the need of create any temporary file.

In the following page, you can see an example of SampleChart.aspx (you can also examine the demo application downloadable from <http://www.softwaresigloxxi.com>)


```

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    Dim oGraph As New Super2d3dGraphLibrary.Super2D3dGraph

    ' First, create a sample 2d Bar chart

    ' Get some random values ...
    Dim arrSeries(1) As ArrayList
    For k As Integer = 0 To UBound(arrSeries)
        arrSeries(k) = New ArrayList
        For j As Integer = 0 To 5
            arrSeries(k).Add(Rnd() * 1000 - 500)
        Next
    Next

    ' Assign properties to Super2d/3dGraphLibrary control ...
    With oGraph

        ' All of the following lines are optional, change them as needed; from here...
        .AutoRefresh = False ' This prevents internal buffer to be painted after each
property changes
        .Series = arrSeries ' Assign previous random values to the chart control
        .Title = "2d Bar Chart" ' Title of the chart
        .SeriesLegend = New String() {"Apples", "Pearls"} ' Names for each serie

        .Style = Super2d3dGraphLibrary.STYLE2D3D.STYLE2D_BAR ' 2d bar classic chart
        .Width = 450 ' Chart width
        .Height = 300 ' Char height
        .BackColor = Color.LightYellow ' Background color
        .BackStyle = Super2d3dGraphLibrary.STYLEBACKGROUND.STYLEBACKGROUND_GRADIENT_INVERTED
        .Calculation = Super2d3dGraphLibrary.CALCULATION.CALCULATION_AVERAGE_SPLINE ' Calculate
an average spline and draw it
        .ValuesLocationStyle = Super2d3dGraphLibrary.STYLECAPTIONLOCATION.STYLECAPTIONLOCATION_INSIDE ' Put values
inside the bars
        .ShowDataTable = False ' Don't need to show the datatable this time
        .CastShadows = True ' Add a subtle shadow to each bar
        .Quality = Drawing2D.SmoothingMode.HighQuality ' Best result
        ' ... to here.

        .RefreshChart() ' Draw the chart on an internal buffer
    End With

    ' Now, let's output the drawn chart...

    ' Change the response headers to output a JPEG image.
    Response.Clear()

```

Super 2d/3d Graph Library Reference

```
Response.ContentType = "image/jpeg"

' Write the image to the response stream in JPEG format.
oGraph.Image.Save(Response.OutputStream, System.Drawing.Imaging.ImageFormat.Jpeg)

' Dispose of the graph control
oGraph.Dispose()
End Sub
```

Populating the Chart and the SeriesFactory helper class

Without SeriesFactory class, you must create an Array of ArrayList (one ArrayList for each serie) and add a value to each of the ArrayList. Then pass the whole variable to Series property.

In order to make easier to populate the chart with values, a helper class is included in the library. Its called "SeriesFactory" class and here's an example of use:

```
Dim helper As New Super2d3dGraphLibrary.SeriesFactory
With helper
    .AddValue(15) ' <-- Add values to the first serie
    .AddValue(25)
    .AddValue(20)
    .NewSerie() ' <-- Begin a new serie, #2
    .AddValue(18)
    .AddValue(23)
    .AddValue(21)
    .ApplyTo(Super2D3dGraph1) ' <-- Assign values to current chart
End With
```

For comparison purposes, here is an example of an straight use of Series property, the traditional way of assigning values to the chart:

```
Dim arrSeries(1) As ArrayList ' <-- 2 series (0 counts).
arrSeries(0) = new ArrayList
arrSeries(0).Add(15)
arrSeries(0).Add(25)
arrSeries(0).Add(20)
arrSeries(1) = new ArrayList
arrSeries(1).Add(18)
arrSeries(1).Add(23)
arrSeries(1).Add(21)
Super2d3dGraphLibrary.Series = arrSeries
```

It's up to you to use either SeriesFactory class or the Series property.

SeriesFactory includes many native methods to make more easy to feed the chart control as well as some special methods to read data from file, such as AddPointsFromFile. For more help and guidance, send your question to our support department.

Tips

1. Speed

To improve the speed of the rendering, set `AutoRefresh` to `False` prior to setting any other property. Then set `AutoRefresh` back to `True` to display the changes in only one paint.

For example:

```
Dim chart as Super2d3dGraphLibrary

With chart
    .AutoRefresh = False
    ' ... Set other properties, like Style, Fonts, Values, etc.
    .AutoRefresh = True
End With
```

2. Output quality

We recommend to use PNG as output format whenever possible (ie. Saving the chart to file or rendering on web applications).

3. Online tutorials

Don't forget to visit us at www.softwaresigloxxi.com from time to time!
We eventually publish new sample code as well as library updates.

Advanced feature: combo charts

Beginning version 8, you can combine two or more charts into a single chart. Super 2d/3d Graph Library will render your main chart, and then, the rest of the suitable elements of the remaining charts in the collection. As a result, now you can have bar/line charts, area/bar/point charts, and so on (only 2d charts are supported by now).

The following sample code shows how to draw a Price/Volume chart (bar/line chart). This sample assumes Super2d3dGraph1 is an existing chart control on a form, then a new chart control is created in memory and added to the main chart.

Step 1: Customize main bar chart and populate with 250 random values. Examine the code and look at fourth line: it's recommended to use Autorefresh=false when assigning properties to the library to improve speed.

```
Dim arrBarChartValues(0) As ArrayList

' Configure Volume bar chart
arrBarChartValues(0) = New ArrayList
With Super2D3dGraph1
    .AutoRefresh = False
    .BackColor = Color.Black
    .WallsBackStyle = STYLEWALLBACKGROUND.STYLEWALLBACKGROUND_TRANSPARENT
    .TitleColor = Color.White
    .Title = "Price/Volume Chart Sample"
    For k As Integer = 0 To 250
        arrBarChartValues(0).Add(Rnd() * 35000)
    Next
    .Series = arrBarChartValues
    .SeriesLegend = New String() {"Volume"}
    .YAxisScaleMode = SCALEMODE.SCALEMODE_FIXED
    .YAxisScaleMaximum = 100000
    .YAxisScaleMinimum = 0
    .YAxisColor = Color.White
    .YAxisTitle = ""
    .Calculation = CALCULATION.CALCULATION_NONE
    .ShowDividerX = False
    .ShowValues = False
    .ShowLegend = True
    .ShowDataTable = False
    .LegendColor = Color.White
    .LegendBackColor = Color.Black
End With
```

Step 2: Create a secondary chart (oLineChart) and customize it. Nothing is drawn yet in the form.

```
' Define Price line chart
Dim oLineChart As New Super2d3dGraphLibrary.Super2D3dGraph
Dim arrSeries(0) As ArrayList, arrItems(250) As String
arrSeries(0) = New ArrayList
Dim p As Integer = Rnd() * 50
With oLineChart
    .AutoRefresh = False
    For k As Integer = 0 To 250
        arrSeries(0).Add(p)
        p += Rnd() * 10 - 5
    Next
End With
```

Super 2d/3d Graph Library Reference

```
        arrItems(k) = k
    Next
    .Series = arrSeries
    .Style = STYLE2D3D.STYLE2D_LINE
    .YAxisLocation = STYLEYAXISLOCATION.STYLEYAXISLOCATION_RIGHT
    .YAxisDividersCount = 10
    .YAxisColor = Color.White
    .SeriesLegend = New String() {"Price"}
    .SeriesLineStyle = New Pen() {New Pen(Color.White)}
    .SeriesColor = New Color() {Color.White}
    .ItemsLegend = arrItems
    .ShowValues = False
    .ShowPoints = False
    .ShowDividerX = True
    .XAxisColor = Color.White
    .XAxisDividersCount = 10
    .XAxisLabelsRotated = True
    .ShowXAxis = True
End With
```

Step 3: Add the line chart to the render engine of the main chart control.

```
' Add the second chart to current chart
Super2D3dGraph1.ChartAdd(oLineChart)
```

Step 4: Now, let Super 2d/3d Graph Library show the results...

```
' Draw the composition
Super2D3dGraph1.RefreshChart()
```

Step 5: Finally, we no longer need the secondary chart, so you can remove it from memory (it's optional). LineChart is second in the chain (first is the main chart), so we pass a 2 to ChartRemove method.

```
' Remove line chart from composite manager
Super2D3dGraph1.ChartRemove(2)
```

Advanced feature: SVG charts (vector charts)

SVG (Scalable Vector Graphics) is supported from version 9.

SVG is different from raster (bitmap) formats because charts in SVG format use a standard XML instead of pixels. This standard XML (the SVG standard) is defined by the w3c. The output is very similar to Adobe Flash. In fact, Adobe has published a SVG plugin for IE.

SVG charts can be resized and transformed in high-quality because it's a vector format.

They also require a "plugin". Firefox and Opera has builtin SVG support. For IE, you can use the free Adobe SVG plugin (you probably have already it installed).

For more information about SVG standard and viewers go to:

<http://www.w3.org/Graphics/SVG/>

http://wiki.svg.org/Viewer_Implementations

Note that at this moment not all chart types can be exported to SVG format. Future versions will increase the number of chart types compatibles. The following chart types are available for SVG export in version 9:

2D chart types supported: area, stacked area, bar area, stacked bar (proportional), line, pie, point, rounded bar, scatter, spline, stacked bar (fixed), horizontal bars, bubble, candlestick, cylinders, high-low, high-low-close, pyramid.

3D chart types supported: stacked area, bar (accumulated), stacked bar, columns (accumulated), stacked columns, line, spline area, spline stacked area.

The following sample code shows how to export the SVG document used to render the chart in the web browser (note that a SVG plugin is required – see above).

First, you need to create a asp.net page that instantiate the library and renders the SVG chart to the browser. Second, you need to put an <embed> tag in the page where you want to draw the chart.

For example, assume that you want to show a SVG chart on the page *webform1.aspx*. In this page, you only need to include an HTML element like this:

```
<embed width="500" height="400" src="StackedBarChart.aspx?.svg"
pluginpage="http://www.adobe.com/svg/viewer/install/" type="image/svg+xml" />
```

Let's examine the above line:

- 1.- Embed tag is preferred since every browser will understand it. You may also use <object> or <iframe> tags.
- 2.- Width and Height are set to fit the browser page. Note that you will need to set also "scaleToFit" parameter later.
- 3.- The "src" argument indicates the name of the aspx page that deals with the library and renders the SVG. Note the strange ?.svg suffix. This is necessary to avoid mime conflicts under IIS or Visual Studio.

Super 2d/3d Graph Library Reference

4.- The "pluginspace" argument indicates the URL where the user may download a SVG viewer plugin if none is installed.

5.- Finally, the "type" argument indicates that the embedded object is a SVG image.

StackedBarChart.aspx will be the page that will instantiate of chart library, calls its members and renders the chart on the Page_load event. This way, chart will be shown on your page without the need of create any temporary file.

In the following page, you can see an example of *StackedBarChart.aspx*.

In the Page_Load event, an object of Super 2d/3d Graph Library is created and populated. Then, a call to CreateSVG method is made to obtain the SVG document that describes the chart.

CreateSVG accepts only one boolean parameter to indicate if the SVG will scale to fit the container or not. Set it to true if you want the SVG to occupy the entire space defined in the <embed> tag.

Next, a few calls to Response object are made to render the content of the SVG to the browser.


```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    Dim oGraph As New Super2d3dGraphLibrary.Super2D3dGraph
    Dim svg As String

    ' First, create a sample chart

    ' Get some random values ...
    Dim arrSeries(1) As ArrayList
    For k As Integer = 0 To UBound(arrSeries)
        arrSeries(k) = New ArrayList
        For j As Integer = 0 To 5
            arrSeries(k).Add(Rnd() * 1000)
        Next
    Next

    ' Assign properties to Super2d/3dGraphLibrary control ...
    With oGraph
        .AutoRefresh = False ' This prevents internal buffer to be painted after each
property changes
        .LicenseIdentifier = "PREM2007"
        .Series = arrSeries ' Assign previous random values to the chart control
        .Title = "Equipment Chart" ' Title of the chart
        .Style = Super2d3dGraphLibrary.STYLE2D3D.STYLE3D_STACKED_BAR
        .CastShadows = True
        .Width = 500 ' Chart width
        .Height = 300 ' Char height
        .BackColor = System.Drawing.Color.LightYellow ' Background color
        .BackStyle = Super2d3dGraphLibrary.STYLEBACKGROUND.STYLEBACKGROUND_GRADIENT_TUBULAR_INVERTED
        .Calculation = Super2d3dGraphLibrary.CALCULATION.CALCULATION_AVERAGE_SPLINE ' Calculate
an average spline and draw it
        .ShowDataTable = False ' Don't need to show the datatable this time
        .ShowLegend = False ' Don't show legend
        .ShowDividerX = False ' Don't show vertical dividers
        .YAxisNumericFormat = "0" ' No decimals on Y-axis
        .ValuesFormat = "0" ' No decimals on values
        .Quality = System.Drawing.Drawing2D.SmoothingMode.HighQuality ' Best result

        ' Get SVG xml document and write to response stream
        svg = oGraph.CreateSVG(False)
    End With

    ' Dispose of the graph control
    oGraph.Dispose()

    ' Change the response headers to output a SVG graph.

```

Super 2d/3d Graph Library Reference

```
Response.Clear()  
Response.ContentType = "image/svg+xml"  
Response.Write(svg)  
Response.End()
```

```
End Sub
```

Advanced feature: PDF export (... and chart printing)

PDF format is supported from version 10.

With `CreatePDF` method you can quickly obtain a PDF representation of the current chart. This method returns a PDF file in a byte array so you can save it to a file, display in a browser in realtime, etc.

`CreatePDF` has the following interface:

```
Public Function CreatePDF(ByVal orientation As STYLE_ORIENTATION, ByVal margin As Integer) As Byte()
```

The resulting PDF is a 100% PDF 1.4 version compatible file, which can be opened with Acrobat Reader so **you can also print the chart from there.**

This method will use current chart image at current resolution (by default, if you don't specify a value with `Resolution` property, image resolution would be 72 dpi – screen resolution). If you want a better image quality for printing purposes you may increase the resolution of the chart just before calling `CreatePDF` method. Example:

```
Imports System.IO
...

Dim buffer as Byte()
With Super2d3dGraph1
    .Resolution = 150
    Buffer = CreatePDF(ORIENTATION_LANDSCAPE, 50)
    File.WriteAllBytes("c:\chart.pdf", buffer)
    .Resolution = 72 'Back to screen resolution
End With
```

Please, **check new samples** of `CreatePDF` usage in our website at <http://www.softwaresigloxxi.com>.

Advanced feature: serialization

Beginning version 10.3, you can save current properties of a chart object (serialization) and recover them later onto a new chart object (deserialization).

Serialization in Super 2d/3d Graph Library is achieved through SettingsFactory inner class which is available with the Settings property.

See online tutorial for usage sample:

http://www.softwaresigloxxi.com/Super2d3dGraphLibrary_Tutorials_Serialization.html

1. To serialize / save all properties to a string variable:

```
super2d3dGraph1.Settings().GetSettings(discardData as Boolean)
```

This will return all properties in a compressed string value. Then you can store the returned string data into a file or database. The parameter discardData allows you to include or exclude current values of chart (replace "super2d3dGraph1" with your chart object name).

2. To deserialize / recover all properties from a string variable:

```
super2d3dGraph1.Settings().Apply(data as String, discardData as Boolean)
```

Pass previous collected string data to "data" parameter and specify if you want to discard any data included in the string data or not (discardData parameter).

Notes:

- Settings included are all non-readonly properties available in Super2d3dGraph class.
- Your activation code (LicenseIdentifier property value) will also be included.

Sample code for Windows application (VB.NET)

All charts require assigning some values to Series properties. The type of Series properties is Array of ArrayList. This property contains all the series and each series contains at least one value. Value can be any numeric expression, although we recommend using Single or Double.

Standard chart types only require one numeric value for each element on the chart (for example, column charts only need one numeric to specify column height). A few chart types, requires two numbers for each value (such as high-low or scatter type; high-low requires a high and a low number for each value; scatter requires one number for X position and another for Y position).

Bubble and high-low-close requires three numbers for each value (in case of bubble, first number defines X position, second number Y position and third number is used to calculate radius of the bubble).

Finally, candlestick charts requires four numbers for each value. They are also known as stock charts and represents open, close, minimum and maximum scores.

The following code assumes that the user control is named oGraph and can be used to draw any chart type of one dimension (one number for each value). Please refer to ASP.NET sample application for more examples of two, three and four numbers for each value.

```
' Get some random values ...
Dim arrSeries(1) As ArrayList
For k As Integer = 0 To UBound(arrSeries)
    arrSeries(k) = New ArrayList
    For j As Integer = 0 To 5
        arrSeries(k).Add(Rnd() * 1000 - 500)
    Next
Next

With oGraph

    ' All of the following lines are optional, change them as needed; from here...
    .AutoRefresh = False                ' This prevents internal buffer to be painted after
each property changes
    .Series = arrSeries                ' Assign previous random values to the chart control
    .Title = "2d Bar Chart"            ' Title of the chart
    .SeriesLegend = New String() {"Apples", "Pearls"} ' Names for each serie

    .Style = Super2d3dGraphLibrary.STYLE2D3D.STYLE2D_BAR ' 2d bar classic chart
    .Width = 450                ' Chart width
    .Height = 300               ' Char height
    .BackColor = Color.LightYellow ' Background color
    .BackStyle =
Super2d3dGraphLibrary.STYLEBACKGROUND.STYLEBACKGROUND_GRADIENT_INVERTED
    .Calculation = Super2d3dGraphLibrary.CALCULATION.CALCULATION_AVERAGE_SPLINE
' Calculate an average spline and draw it
    .ValuesLocationStyle =
Super2d3dGraphLibrary.STYLECAPTIONLOCATION.STYLECAPTIONLOCATION_INSIDE ' Put values
inside the bars
```

Super 2d/3d Graph Library Reference

```
.ShowDataTable = False           ' Don't need to show the datatable this time
.CastShadows = True              ' Add a subtle shadow to each bar
.Quality = Drawing2D.SmoothingMode.HighQuality    ' Best result
' ... to here.

.RefreshChart()                  ' Draw the chart on an internal buffer
End UIT
```

Sample code for C# (Windows application)

All charts require assigning some values to Series properties. The type of Series properties is Array of ArrayList. This property contains all the series and each series contains at least one value. Value can be any numeric expression, although we recommend using Single or Double.

Standard chart types only require one numeric value for each element on the chart (for example, column charts only need one numeric to specify column height). A few chart types, requires two numbers for each value (such as high-low or scatter type; high-low requires a high and a low number for each value; scatter requires one number for X position and another for Y position).

Bubble and high-low-close requires three numbers for each value (in case of bubble, first number defines X position, second number Y position and third number is used to calculate radius of the bubble).

Finally, candlestick charts requires four numbers for each value. They are also known as stock charts and represents open, close, minimum and maximum scores.

The following code illustrates how to assign values to Series property:

```
ArrayList[] data = new ArrayList[2];
Random rnd = new Random();

for (int serie=0;serie<2;serie++)
{
    data[serie] = new ArrayList();    // Each ArrayList corresponds to a "serie"
    for(int k =0;k<5;k++)
    {
        // Put two values for each "serie"
        data[serie].Add ( rnd.Next(-100,100) );
        data[serie].Add ( rnd.Next(100) );
    }
}
super2D3dGraph1.Series = data;
super2D3dGraph1.RefreshChart();
```

Sample code: creating charts without using forms (C#)

It's very easy to render a chart using only code and no forms (useful for a Windows service or if you don't want to use the chart as an user control for example):

```
Super2d3dGraphLibrary.Super2D3dGraph oGraph = new Super2d3dGraphLibrary.Super2D3dGraph();
```

```
ArrayList[] data = new ArrayList[2];  
Random rnd = new Random();
```

```
for (int serie=0;serie<2;serie++)  
{  
    data[serie] = new ArrayList(); // Each ArrayList corresponds to a "serie"  
    for(int k =0;k<5;k++)  
    {  
        // Put two values for each "serie"  
        data[serie].Add ( rnd.Next(-100,100) );  
        data[serie].Add ( rnd.Next(100) );  
    }  
}  
oGraph.Width = 700;  
oGraph.Height = 400;  
oGraph.Series = data;  
oGraph.Image.Save("c:\\test.png", System.Drawing.Imaging.ImageFormat.Png);
```


Library Quick Reference

Properties

Version	Member	Type	Description	Default value
2	AutoRefresh	Boolean	Set it chart is refreshed after any change to properties. If set to false, you need to call RefreshChart to manually refresh of chart.	True
3	BackColor	System. Drawing.Color	Chart's background color	Color.White
3	BackStyle	STYLEBACKGROUND	Chart's background style. Can be any of the following: STYLEBACKGROUND_SOLID = 0 STYLEBACKGROUND_GRADIENT = 1 STYLEBACKGROUND_GRADIENT_INVERTED = 2	STYLEBACKGROUND_SOLID
1	BorderColor	System. Drawing.Color	Chart's border color	Color.Red
5	BubbleMaxRadius	Long	Maximum size of a bubble (bubble chart only)	100
7	BuiltinDialogAllowed	Boolean	Enable or disable right-button contextual menú	True
7	BuiltinDialogLanguage	LANGUAGE	Contextual menú language	LANGUAGE.LANG_ENGLISH
3	Calculation	CALCULATION	Additional calculation to draw on chart: CALCULATION_NONE = 0 CALCULATION_AVERAGE = 10 CALCULATION_AVERAGE_SPLINE = 11 CALCULATION_PARETO = 20	CALCULATION_NONE
3	CalculationLineStyle	Pen	Line style for additional calculation	New Pen(Color.Blue, 3)
12	CalculationTrendLine	TrendLine	Returns a TrendLine object containing the parameters of the line (slope and y-displacement)	Nothing
1	CastShadows	Boolean	Show/hide a shadow	False
12.5	DataSource	(DataGridView control)	A DataGridView control to databind	Nothing
12.5	DataSourceMode	DATASOURCE_SCHEME	Specifies how the data in datasource must be interpreted	Nothing
12.5	DataTableAdditionalRowTitle	String	Title for an additional fixed row of custom values	""
15.5	DataTableAdditionalRowValues	String()	Values of an additional and optional row of custom values	Nothing
5	DataTableBackColor	Color	Datatable's background color	Color.White
5	DataTableColor	Color	Datatable's text color	Color.Black
9	DataTableFont	Font	Datatable's text font	New Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point)
12	DataTableLastIndex	Integer	Index of last value to be displayed in the datatable	(Last item)
12	DataTableStartIndex	Integer	Index of first value to be displayed in the	1

Super 2d/3d Graph Library Reference

			datatable	
4	DataTableText	String	Datatable's caption	"Datatable"
6.5	Depth	Integer	Thickness of 3d pie or 3c doughnut charts	30
12.5	ShowFrontLayer	Boolean	Show/hide a front solid layer which can be partially transparent	False
12.5	FrontLayerColor	System.Drawing.Color	Color for the front layer	Nothing
10.1	GanttSetToday	DateTime	Position for the "today" line in a Gantt chart	Now
10.1	GanttSetTodayLabel	String	Text to draw at the "today" line	"Today"
9.5	GaugeBackStyle	SolidBrush	Gauge's background style	Nothing
9.5	GaugeBigTickStyle	Pen	Gauge's big tick pen style	Nothing
9.5	GaugeBorderStyle	Pen	Gauge's border style	Nothing
9.5	GaugeCentralCircleRadius	Single	Gauge's central circle radius	0
9.5	GaugeCentralCircleBackStyle	SolidBrush	Gauge's central circle background style	Nothing
9.5	GaugeCentralCircleBorderStyle	Pen	Gauge's central circle border style	Nothing
9.5	GaugeMarginFactor	Single	Image to be rendered on the grid's background	0.1
9.5	GaugeNeedleStyle	Pen	Gauge's needle pen style	Nothing
12	GaugeRangeColors	ArrayList()	For each serie can specify intervals which are painted with custom color. The array length must match the number of series and in each ArrayList you may specify any number of intervals in the form of (low, high, color).	Nothing
9.5	GaugeSmallTickStyle	Pen	Gauge's small tick pen style	Nothing
12	GaugeTitles	String ()	Each gauge's top title	Nothing
12	GaugeTitlesFont	Font	Font style for all gauges' titles	New Font("Verdana", 10)
6	GridBackgroundImage	Image	Image to be rendered on the grid's background	Nothing
6	GridBackgroundImage Transparency	Integer	Transparency percentage factor (0= fully visible, 100=transparent)	0
10.4	GridLineStyle	Pen	Grid line style and color	Pen(Color.DarkGray)
6.5	HoleSize	Integer	Size of central hole in doughnut charts	30
9	HotSpotBackColor	Color	Background color of value tooltips	Color.FromArgb(220, 255, 255, 64)
9	HotSpotColor	Color	Text color of value tooltips	Color.FromArgb(255, 0, 0, 128)
10	HotSpotCrossLineStyle	Pen	Line style of tooltip cross (see HotSpotShowCross)	Pen(Color.FromArgb(255, 220, 32, 32))
10	HotSpotDateTimeFormat	String	Text format when tooltip shows a date/time value	"d"
9	HotSpotsEnabled	Boolean	Activate/deactivate hotspots (tooltips)	"N"
10	HotSpotNumericFormat	String	Numeric format when tooltip shows a numeric value	"N"
12.5	HotSpotOutlineEnabled	Boolean	Allows user selection of chart items using mouse	True
12.5	HotSpotOutlineStyle	Pen	Specifies outline style for items being selected	New Pen(Color.Yellow)
10.2	HotSpotTooltips	ArrayList()	Allows manual setting of each item tooltip	Nothing (automatic)

Super 2d/3d Graph Library Reference

10	HotSpotShowCross	Boolean	Show/hide a cross over the tooltip	False
1	Image	Bitmap	Current chart bitmap	
6	ItemsHighLight	String()	Assign any text to any item to explode pie chart. For example, ItemsHighlight(3) = "explode" would cause item #3 to separate in a pie chart. Now you can assign any text. In future versions, this text could be shown over each exploded sector of the pie chart describing the cause it is "highlighted"	Empty
2	ItemsLegend	String()	X axis items names	Empty
6.5	LegendAlignment	STYLEALIGNMENT	Position for legend	STYLEALIGNMENT_MIDDLE_RIGHT
8	LegendBackColor	Color	Legend's back color	Color.White
2	LegendColor	Color	Legend's text color	Color.Black
9	LegendFont	Font	Legend's text font	New Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point)
4	LegendText	String	Legend's caption	"Legend"
5	LicenseeIdentifier	Long	Unique identity number for licensed user	0
5.5	MarkHighest	Boolean	Draws label for highest value in yellow background and blue text	False
5.5	MarkLowest	Boolean	Draws label for lowest value in red background and yellow text	False
7.5	MarginBottom	Integer	Inner distance in pixels between lower limit of the chart and the border	20
7.5	MarginLeft	Integer	Inner distance in pixels between left limit of the chart and the border	20
7.5	MarginRight	Integer	Inner distance in pixels between right limit of the chart and the border	20
7.5	MarginTop	Integer	Inner distance in pixels between top limit of the chart and the border	20
10.6	PieBorderStyle	Pen	Line style for the borders of pie segmengs	New Pen(Color.Transparent)
5	PointStyle	STYLEPOINT	Point style (point and scatter charts)	STYLEPOINT_CIRCLE
5	PointSize	Integer	Point size (point and scatter charts)	4
2	Quality	System.Drawing.Drawing2d.SmoothingMode	Render quality	Default
1	Resolution	Long	Bitmap resolution (ppp)	72
1	Series	ArrayList	List of arrays containing the values for each serie. For pie chart, only first serie is taken in account.	(2 series with eight random values each)
10	SeriesBorderStyle	Array of System.Drawing.Pen	Contains a pen object for each serie's bar borders	
1	SeriesColor	Array of System.Drawing.Color	Array with each series color. For pie chart, each sector colors correspond to each value on the first	Random

Super 2d/3d Graph Library Reference

			serie	
1	SeriesFillStyle	Array of System.Drawing.Brush	Contains a brush object for each serie (only for Line chart type)	Nothing (automatic)
11.6	SeriesGap	Single()	Specify distance from each serie in percentage (i.e. 0.5) or fixed amount (i.e. 40 pixels)	0.25
1	SeriesLegend	Array of String	Series names	
1	SeriesLineStyle	Array of System.Drawing.Pen	Contains a pen object for each serie (only for Line chart type)	Nothing (automatic)
6	SeriesPicture	Array of Image	Contains an image object for each serie of the chart. Must be used in picture charts.	Nothing
11.1	SeriesPointSize	Integer()	Allows setting individual series' point size	Nothing (automatic)
11.1	SeriesPointStyle	STYLEPOINT()	Allows setting individual series' point style	Nothing (automatic)
10.2	SettingsFromString	String	Apply look and feel settings generated with Chart wizard tool or with Settings.GetSettings function	
8.5	ShadowColored	Boolean	Draw shadows in color or in gray	False (in gray)
8.5	ShadowSmooth	Boolean	Use diffuse shadows	True
8.5	ShadowOffset	Integer	Distance of the shadow in pixels	5
1	ShowDataTable	Boolean	Show/hide data table	True
1	ShowDividerX	Boolean	Show/hide vertical separators	True
1	ShowDividerZ	Boolean	Show/hide depth separators	True
2	ShowItemsLegend	Boolean	Show/hide X axis items names	False
1	ShowLegend	Boolean	Show/hide items legend	True
9	ShowLegendInDataTable	Boolean	Show/hide series legend in data table	True
6.5	ShowPoints	Boolean	Whether additional marks or points are shown over line, spline or area charts	True
1	ShowSolidColors	Boolean	Set to true if you don't want gradients on fills	False
2	ShowSurface3d	Boolean	Show/hide surface line on 3d charts	False
1	ShowValues	Boolean	Show/hide values on chart	True
10.5	ShowValuesMax	Double	Max value to be displayed	Double.Max (unlimited)
10.5	ShowValuesMin	Double	Min value to be displayed	Double.Min (unlimited)
3	ShowXAxis	Boolean	Show/hide values over X Axis	True
11	ShowYAxis	Boolean	Show/hide values over Y Axis	True
11	ShowZAxis	Boolean	Show/hide values over Z Axis	True
2	SlantX	Single	X axis offset for 3d effect	30
2	SlantY	Single	Y axis offset for 3d effect	30
1	Style	STYLE2D3D	Chart's style	STYLE2D_BAR
11.1	SurfaceBrushMaxBrightness	Integer	In Surface charts, the brightness of the peak	100
11.1	SurfaceBrushMinBrightness	Integer	In Surface charts, the brightness of the valley	0
11.1	SurfaceBrushStyle	Brush	In Surface charts, the brush of the terrain	SolidBrush(Color.FromArgb(128, 128, 128, 128))
12.5	SurfaceWireOnly	Boolean	In Surface charts, only renders a wireframe	False

Super 2d/3d Graph Library Reference

1	Style	STYLE2D3D	Chart's style	STYLE2D_BAR
7	Thickness	Integer	3D cylinders' width	30
1	Title	String	Chart's title	"Title"
1	TitleFont	System. Drawing.Font	Chart title's font	
1	TitleColor	System. Drawing.Color	Chart title's text colour	Color.Black
9	ValuesAlignment	StringAlignment	Alignment for values in the datatable	StringAlignment.Far
2	ValuesBackColor	System. Drawing.Color	Background colour for values	Color.Orange
11	ValuesCaptions	String()	Allows specify individual item's captions	Nothing (automatic)
2	ValuesColor	System. Drawing.Color	Text colour for values	Color.Black
2	ValuesFont	System. Drawing.Font	Values' font	
2	ValuesFormat	String	String format for values	"N"
3	ValuesLocationStyle	STYLECAPTIONLOCATION	Location style for values	STYLECAPTIONLOCATION_DEFAULT
2	ValuesStyle	STYLECAPTION	Appearance style for values	STYLECAPTION_ROUNDED_SQUARE
5	WallsBackStyle	STYLEWALLBACKGROUND	Fill style for grid	STYLEWALLBACKGROUND_BANDS
5	WallsBackColor Principal	Color	Main colour for grid background	Color.FromArgb(64, 148, 232, 255)
5	WallsBackColor Secondary	Color	Secondary colour for grid background	Color.FromArgb(128, 148, 232, 255)
2	WallStyle	STYLEWALLS	Style for walls (only 3d charts)	STYLEWALLS_NOWALLS
11	XAxisAngle	Single	Rotation angle for X-Axis (only XYZ charts)	PI / 10
8	XAxisColor	Color	Color for X-axis texts	Color.Black
8	XAxisDividerCount	Integer	Maximum number of items in the X-Axis	0 (all items are shown by default)
8	XAxisHeight	Integer	Height in pixels for the space allocated for X-Axis labels	1000
5.5	XAxisLabelsRotated	Boolean	Draws X-axis labels rotated 45°	False
8.5	XAxisLabelsRotated90	Boolean	Draws X-axis labels rotated 90°	False
6	XAxisNumericFormat	String	Expression format for X-axis values. Use "0" for integer values only.	"N"
6	XAxisScaleMode	SCALEMODE	Scale mode for X-axis (automatic or fixed). If fixed, use XAxisScaleMinimum and XAxisScaleMaximum to set the scale range.	SCALEMODE.AUTOMATIC
6	XAxisScaleMaximum	Single	Maximum value for X-axis scale	0
6	XAxisScaleMaximumDateTime	DateTime	Maximum value for X-axis scale in DateTime mode	
6	XAxisScaleMinimum	Single	Minimum value for X-axis scale	0
6	XAxisScaleMinimumDateTime	DateTime	Minimum value for X-axis scale in DateTime mode	
11	XAxisTickLength	Integer	Length in pixels of X-axis ticks	5
6.5	XAxisTitle	String	X-Axis title	""
6.5	XAxisTitleFont	Font	X-Axis title's font	Font("Arial", 11, FontStyle.Bold, GraphicsUnit.Point)
9	XAxisValuesFont	Font	X-Axis labels' font	Font("Arial", 12, FontStyle.Bold, GraphicsUnit.Point)
11	YAxisAngle	Single	Rotation angle for Y-Axis (only XYZ charts)	PI / 5

Super 2d/3d Graph Library Reference

8	YAxisColor	Color	Color for Y-Axis texts	Color.Black
1	YAxisDividersCount	Long	Number of Y axis dividers. If set to 10, you can emulate a percentage axis. If set to zero, no divider is shown.	10
9.5	YAxisIndicators	Array of Y_AXIS_INDICATORS	Array of Y_AXIS_INDICATORS structure. Each indicator draws an horizontal line over the chart useful for marking special limits.	Nothing
5.5	YAxisLabelsRotated	Boolean	Draws Y-axis labels rotated 45°	False
8	YAxisLocation	STYLEYAXISLOCATION	Location for Y-Axis (left or right)	STYLEYAXISLOCATION_LEFT
6	YAxisNumericFormat	String	Expression format for Y-axis values. Use "0" for integer values only.	"N"
6	YAxisScaleMode	SCALEMODE	Scale mode for y-axis (automatic or fixed). If fixed, use YAxisScaleMinimum and YAxisScaleMaximum to set the scale range.	SCALEMODE.AUTOMATIC
6	YAxisScaleMaximum	Single	Maximum value for y-axis scale	0
6	YAxisScaleMinimum	Single	Minimum value for y-axis scale	0
10	YAxisTickLength	Integer	Length in pixels of Y-axis ticks	5
6.5	YAxisTitle	String	Y-Axis title	""
6.5	YAxisTitleFont	Font	Y-Axis title's font	Font("Arial", 11, FontStyle.Bold, GraphicsUnit.Point)
9	YAxisValuesFont	Font	Y-Axis labels' font	Font("Arial", 12, FontStyle.Bold, GraphicsUnit.Point)
11	ZAxisAngle	Single	Rotation angle for Z-Axis (only XYZ charts)	PI / 5
11	ZAxisColor	Color	Color for Z-Axis texts	Color.Black
11	ZAxisDividersCount	Long	Number of Z axis dividers. If set to 10, you can emulate a percentage axis. If set to zero, no divider is shown.	10
11	ZAxisNumericFormat	String	Expression format for Y-axis values. Use "0" for integer values only.	"N"
11	ZAxisScaleMode	SCALEMODE	Scale mode for y-axis (automatic or fixed). If fixed, use ZAxisScaleMinimum and ZAxisScaleMaximum to set the scale range.	SCALEMODE.AUTOMATIC
11	ZAxisScaleMaximum	Single	Maximum value for y-axis scale	0
11	ZAxisScaleMinimum	Single	Minimum value for y-axis scale	0
11	ZAxisTickLength	Integer	Length in pixels of Y-axis ticks	5
11	ZAxisTitle	String	Y-Axis title	""
11	ZAxisTitleFont	Font	Y-Axis title's font	Font("Arial", 11, FontStyle.Bold, GraphicsUnit.Point)
11	ZAxisValuesFont	Font	Y-Axis labels' font	Font("Arial", 12, FontStyle.Bold, GraphicsUnit.Point)

Methods

Version	Member	Parameters	Description
2	Animate	None	Animate the chart beginning with zero for all values to original values.
8	ChartAddNew	None	Adds a new initialized chart to the render engine
8	ChartAddNew	ByVal oChart As Super2d3dGraphLibrary.Super2D3dGraph	Adds a chart object to the render engine
8	ChartCount	Integer	Returns the number of chart instances
8	ChartItem	ByVal item As Integer	Returns a reference to the chart object specified by Item (1, 2, 3..)
8	ChartRemove	ByVal item As Integer	Removes from chart engine the chart object specified by item
8.5	CreateAnimatedGIF	ByVal intFrames As Integer, ByVal intSpeed As Integer, ByVal blnLoop As Boolean, ByVal style As STYLEANIMATION	Returns an animated GIF image
10	CreatePDF	ByVal orientation As STYLE_ORIENTATION, ByVal margin As Integer	Returns a PDF representation of current chart
9	CreateSVG	ByVal scaleToFit As boolean	True will scale the chart to fit the entire space defined in the holding tag (embed, object or iframe)
2	RefreshChart	None	Force a chart update.
10.3	Reset	None	Clears all data and returns all properties to default values
12	SetAllSeriesLineStyle	Pen p	Assigns the same line style (p) to all series

Events

Version	Nombre	Parameters	Description
3	BeginPaint	g as graphics	Chart is cleared and before anything is drawn
3	BeforeEndPaint	G as graphics	Just before image is updated on the form
3	AfterPaint	G as graphics	After image is updated on the form
11	ValueClick	ByVal g As Graphics, ByVal serie As Integer, ByVal itemIndex As Integer	Occurs when user click over an item (bar, point, ...)

Library Reference

Animate

Description

Start chart animation. Animation style depends on parameters specified.

Definition

```
Public Sub Animate(ByVal style As STYLEANIMATION, ByVal stepping As Integer)
```

Parameters

style	Can be one of the following: STYLEANIMATION_RISE = 1 STYLEANIMATION_WAVE = 2
stepping	Animation speed, from 1 (slowest) to 100 (fastest)

Autorefresh

Description

Set control's refreshing mode. True = automatic refresh, False = manual refresh.

- Automatic (default): control will be repainted after any method call that could change its appearance, such as `BorderColor`.
- Manual: control will not be repainted until a call to `RefreshChart`. Setting `Autorefresh` to automatic will also repaint the control.

Definition

`Property Autorefresh() As Boolean`

Parameters

Boolean True (automatic) or False (manual).

Returns

Current state of control's autorefreshing mode (true or false).

BackColor

Description

Set control's background color. Note that this property only changes the base color for the background. The final appearance of background depends on BackStyle property.

Default value is White.

Definition

```
Public Overrides Property BackColor() As System.Drawing.Color
```

Parameters

System.Drawing.Color static function).	Any color may be used (you can create any color using Color.FromArgb
---	--

Returns

Current background color.

BackStyle

Description

Set control's background style: solid or gradient fill. If you choose a gradient fill, the initial and end colors of the gradient will be calculated using the base color set with BackColor property.

Default value is STYLEBACKGROUND_SOLID.

Definition

```
Public Property BackStyle() As STYLEBACKGROUND
```

Parameters

STYLEBACKGROUND

Can be any of the following values:

```
STYLEBACKGROUND_SOLID = 0  
STYLEBACKGROUND_GRADIENT = 1  
STYLEBACKGROUND_GRADIENT_INVERTED = 2  
STYLEBACKGROUND_GRADIENT_TUBULAR = 3  
STYLEBACKGROUND_GRADIENT_TUBULAR_INVERTED = 4
```

Returns

Current style for background.

BorderColor

Description

Set control's border color

Default value is Color.Red.

Definition

`Property` BorderColor() `As` Color

Parameters

Color Any color may be used (you can create any color using Color.FromArgb static function).

Returns

Current color of the control's border.

BubbleMaxRadius

Description

Specify the maximum diameter of bubbles (only applies to Bubble Chart).
The diameter of any one bubble is calculated taking the maximum diameter and multiplying it by value/maximum value.

Default value is 100 pixels.

Definition

`Property BubbleMaxRadius() As Integer`

Parameters

Integer	Maximum diameter of the bubbles in pixels.
---------	--

Returns

Current maximum diameter of the bubbles in pixels.

BuiltinDialogAllowed

Description

Set if contextual menu is shown when user right-clicks the chart control.

Default value is TRUE.

Definition

`Property BuiltinDialogAllowed() As Boolean`

Parameters

Boolean	Allow or not allow contextual menu
---------	------------------------------------

Returns

Current contextual menu preference (allowed or not allowed).

BuiltinDialogLanguage

Description

Set contextual menu language.

Default value is LANGUAGE.LANG_ENGLISH

Definition

`Property BuiltinDialogLanguage() As LANGUAGE`

Parameters

LANGUAGE	May be one of the following values: LANGUAGE.LANG_ENGLISH LANGUAGE.LANG_SPANISH
----------	---

Returns

Current contextual menu language preference.

Calculation

Description

Performs an additional calculation and draw it over the chart. Setting this property you can easily add an average spline or a line that links the top of the bars. The calculation only is applied to some type of charts, like bars, cylinder, pyramid and point chart.

Default value is CALCULATION_NONE

Definition

```
Property Calculation() As Calculation
```

Parameters

CALCULATION	May be one of the following values: CALCULATION_NONE = 0 CALCULATION_AVERAGE = 10 CALCULATION_AVERAGE_SPLINE = 11 CALCULATION_PARETO = 20 CALCULATION_TOP_LINE = 30 CALCULATION_TOP_SPLINE = 31
-------------	---

Returns

Current calculation preference.

CalculationLineStyle

Description

Specify the line style of the additional calculation.

Default value is Pen(Color.Blue, 3)

Definition

`Property CalculationLineStyle() As Pen`

Parameters

Pen Any System.Drawing.Pen object.

Returns

Pen object holding the current calculation line style

CastShadows

Description

Draw a subtle shadow behind the graph elements or not.

Default value is False.

Definition

`Property CastShadows() As Boolean`

Parameters

Boolean	Either a subtle shadow is drawn behind the graph elements or not.
---------	---

Returns

Current preference.

ChartAddNew

Description

Adds a new, initialized chart object to the render engine of another chart. This method is used to combine two or more charts into a single chart. No changes are visible until RefreshChart is called.

See Advanced Features at the beginning of this guide for more information.

Definition

```
Public Sub ChartAddNew()
```

Parameters

None.

Returns

Nothing.

ChartAdd

Description

Adds an previously created chart object to the render engine of another chart. This method is used to combine two or more charts into a single chart. No changes are visible until RefreshChart is called.

See Advanced Features at the beginning of this guide for more information.

Definition

```
Public Sub ChartAdd(ByVal oChart As Super2d3dGraphLibrary.Super2D3dGraph)
```

Parameters

oChart	Chart to be added to the main chart
--------	-------------------------------------

Returns

Nothing.

ChartItem

Description

Returns a chart object identified by index.

Definition

```
Public Function ChartItem(ByVal index As Integer) As  
Super2d3dGraphLibrary.Super2D3dGraph
```

Parameters

index	Integer index (1=main chart, 2 and above returns additional charts previously added with ChartAdd or ChartAddNew)
-------	---

Returns

Super2d3dGraph object

ChartRemove

Description

Removes an existing chart object in the render engine of another chart identified by index.

Definition

```
Public Sub ChartRemove(ByVal index As Integer)
```

Parameters

index	Integer index (MUST be 2 or greater)
-------	--------------------------------------

Returns

Nothing

CreateAnimatedGIF

Description

Returns an image object containing an animated GIF image.

Definition

```
Public Function CreateAnimatedGIF(ByVal intFrames As Integer, ByVal intSpeed As Integer,
ByVal blnLoop As Boolean, ByVal style As STYLEANIMATION) As Image
```

Parameters

intFrames	Number of frames
intSpeed	Delay in ms between frames
blnLoop	True -> repeat animation forever
style	Type of animation. Can be one of the following values: STYLEANIMATION_RISE = 1 STYLEANIMATION_WAVE = 2

Returns

Image object.

Note: you can use Image.Save method to export it to a disk file for example:

CreatePDF

Description

Returns a PDF representation of current chart.

Definition

```
Public Function CreatePDF(ByVal orientation As STYLE_ORIENTATION, ByVal margin As Integer) As Byte()
```

Parameters

orientation	Can be one of the following: ORIENTATION_PORTRAIT = 0 ORIENTATION_LANDSCAPE = 1
Margin	Amount of margin in points (eg: 50)

Returns

PDF containing chart image.

Note: see chapter "Advanced feature: PDF Export" for more information.

CreateSVG

Description

Returns an xml document in SVG format describing the current chart.

Definition

```
Public Function CreateSVG(ByVal scaleToFit As Boolean) As String
```

Parameters

scaleToFit	True -> scale to fit canvas
------------	-----------------------------

Returns

SVG xml document.

Note: see chapter "Advanced feature: SVG charts" for more information.

DataSource

Description

Allows databinding to another DataGridView control.
You must also set DataSourceMode to help chart control understand the underlying data structure.
Please, refer to online tutorials for databinding samples (XYZ chart databind tutorial for example).

Definition

`Property DataSource() As DataGridView`

Parameters

DataGridView	The control to which databind chart.
--------------	--------------------------------------

Returns

Reference to current databound control

DataSourceMode

Description

Helps chart control understand the underlying data structure of the databound control.

Definition

```
Public Property DataSourceMode() As DATASOURCE_SCHEME
```

Parameters

DATASOURCE_SCHEME	Structure of underlying data.
-------------------	-------------------------------

Must be one of the following values:

NONE = 0

Value when databinding is not enabled

XZ_HEADERS_Y_CELLS = 1

Only for XYZ charts. In this case, the X axis range is determined by column headers values and the Z axis range is determined by row headers values. Y values are the cells' values.

CELL_VALUES = 2

Only for 1D charts (bar, pie, ...). Each row corresponds to a serie. Each value/point equals to a cell in that row.

Returns

Reference to current databound control

DataTableBackColor

Description

Sets datatable's background color.

Default value is Color.White.

Definition

`Property DataTableBackColor() As Color`

Parameters

Color Any color may be used (you can create any color using Color.FromArgb static function).

Returns

Current datatable's background color.

DataTableColor

Description

Sets datatable's text color.

Default value is Color.Black.

Definition

`Property DataTableColor() As Color`

Parameters

Color Any color may be used (you can create any color using Color.FromArgb static function).

Returns

Current datatable's text color.

DataTableFont

Description

Sets datatable's text font.

Default value is `New Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point).`

Definition

```
Property DataTableFont () As Font
```

Parameters

Font	Any font object may be used.
------	------------------------------

Returns

Current datatable's text font.

DataTableText

Description

Specify datatable's title.

Default value is "Datatable".

Definition

```
Property DataTableText() As String
```

Parameters

String	Title or caption for datatable.
--------	---------------------------------

Returns

Current datatable's title.

Depth

Description

Specify thickness (vertical size) for 3D pie and doughnut charts.

Default value is 30 pixels.

Definition

```
Public Property Depth() As Integer
```

Parameters

Integer	Vertical size of pies or doughnut charts
---------	--

Returns

Current size in pixels.

GanttSetToday

Description

Set or retrieves the date assigned to the "Today" vertical line in a Gantt chart.

Default value is Now.

Definition

```
Public Property GanttSetToday() As DateTime
```

Parameters

DateTime	Instant value for the "today" line position
----------	---

Returns

Current value.

GanttSetTodayLabel

Description

Set or retrieves the text assigned to the "Today" vertical line in a Gantt chart.

Default value is "Today".

Definition

```
Public Property GanttSetTodayLabel() As String
```

Parameters

String	Text for the "today" line position
--------	------------------------------------

Returns

Current value.

GaugeBackStyle

Description

Array of SolidBrush objects for background style of each gauge. Note that you can draw several gauges at once.

Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeBackStyle() As SolidBrush()
```

Parameters

SolidBrush()	Array of SolidBrush objects.
--------------	------------------------------

Returns

Current values.

GaugeBigTickStyle

Description

Array of Pen objects for big ticks of each gauge. Note that you can draw several gauges at once.

Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeBigTickStyle() As Pen()
```

Parameters

Pen() Array of Pen objects.

Returns

Current values.

GaugeBorderStyle

Description

Array of Pen objects for border style of each gauge. Note that you can draw several gauges at once.
Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeBorderStyle() As Pen()
```

Parameters

Pen() Array of Pen objects.

Returns

Current values.

GaugeCentralCircleBackStyle

Description

Array of SolidBrush objects for the central circle background style of each gauge. Note that you can draw several gauges at once.

Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeCentralCircleBackStyle() As SolidBrush()
```

Parameters

SolidBrush()	Array of SolidBrush objects.
--------------	------------------------------

Returns

Current values.

GaugeCentralCircleBorderStyle

Description

Array of Pen objects for the central circle border style of each gauge. Note that you can draw several gauges at once.

Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeCentralCircleBorderStyle() As Pen()
```

Parameters

Pen() Array of Pen objects.

Returns

Current values.

GaugeCentralCircleRadius

Description

Array of Single values for the radius of each gauge. Note that you can draw several gauges at once.
Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeCentralCircleRadius() As Single()
```

Parameters

Single()	Array of Single values.
----------	-------------------------

Returns

Current values.

GaugeCentralCircleRadius

Description

Array of Single values for the radius of each gauge. Note that you can draw several gauges at once. Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeCentralCircleRadius() As Single()
```

Parameters

Single()	Array of Single values.
----------	-------------------------

Returns

Current values.

GaugeMarginFactor

Description

Percentage factor for inner margin between each gauge.

Default value is 0.1. You may set a higher or lower value than 0.1.

Definition

```
Public Property GaugeMarginFactor() As Single
```

Parameters

Single Decimal value that's the percentage of separation among each gauge graph.

Returns

Current values.

GaugeNeedleStyle

Description

Array of Pen objects for each gauge's central needle. Note that you can draw several gauges at once. Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeNeedleStyle() As Pen()
```

Parameters

Pen() Array of Pen objects.

Returns

Current values.

GaugeSmallTickStyle

Description

Array of Pen objects for small ticks of each gauge. Note that you can draw several gauges at once. Default value is Nothing, so it's mandatory that you set this property prior to draw gauge charts.

Definition

```
Public Property GaugeSmallTickStyle() As Pen()
```

Parameters

Pen() Array of Pen objects.

Returns

Current values.

GridBackgroundImage

Description

Assign an image object to grid's background.

Default value is Nothing.

Definition

```
Public Property GridBackgroundImage() As Image
```

Parameters

Image Image object to be set as grid's background.

Returns

Current background image object.

GridBackgroundImageTransparency

Description

Set a level of transparency to grid background image.

Default value is 0 (opaque).

Definition

```
Public Property GridBackgroundImageTransparency() As Integer
```

Parameters

Integer	% of transparency (0=opaque, 100=fully transparent)
---------	---

Returns

Current transparency level.

GridLineStyle

Description

Set the line style and color for grid's vertical and horizontal lines.
Default value is Pen(Color.DarkGray).

Definition

```
Public Property GridLineStyle() As Pen
```

Parameters

Pen	Pen object with color and line style
-----	--------------------------------------

Returns

Current line style and color (pen object)

HoleSize

Description

Specify hole size in pixels for doughnut charts.

Default value is 30 pixels.

Definition

```
Property HoleSize() As Integer
```

Parameters

Integer	Hole size in pixels
---------	---------------------

Returns

Current hole size in pixels.

HotSpotBackColor

Description

Specify background color for hotspot (tooltips shown when mouse moves over a value).

Default value is `Color.FromArgb(220, 255, 255, 64)`

Definition

```
Public Property HotSpotBackColor() As Color
```

Parameters

Color Background color. Use `Color.FromArgb` to create a new color.

Returns

Current setting.

HotSpotBackColor

Description

Specify background color for hotspots (tooltips shown when mouse moves over a value).

Default value is `Color.FromArgb(220, 255, 255, 64)`

Definition

```
Public Property HotSpotBackColor() As Color
```

Parameters

Color Background color. Use `Color.FromArgb` to create a new color.

Returns

Current setting.

HotSpotColor

Description

Specify text color for hotspots (tooltips shown when mouse moves over a value).

Default value is `Color.FromArgb(255, 0, 0, 128)`

Definition

```
Public Property HotSpotColor() As Color
```

Parameters

Color Text forecolor. Use `Color.FromArgb` to create a new color.

Returns

Current setting.

HotSpotsCrossLineStyle

Description

Set or returns current line style of tooltip cross. See also HotSpotShowCross property.

Default value is `Pen(Color.FromArgb(255, 220, 32, 32))`.

Definition

```
Public Property HotSpotCrossLineStyle() As Pen
```

Parameters

Pen	Any pen object specifying color and/or line style.
-----	--

Returns

Current setting.

HotSpotsEnabled

Description

Set whether values tooltips will be shown when mouse is moved over the chart.

Default value is False

Definition

```
Public Property HotSpotsEnabled() As Boolean
```

Parameters

Boolean	Enable/disable tooltips
---------	-------------------------

Returns

Current setting.

HotSpotsDateTimeFormat

Description

Set or returns datetime format for tooltips when a date/time is shown. Only works in charts that have a datetime range in the x-axis, like X/Y DateTime, X/Y Line DateTime or any chart which x-axis is set to FIXED_DATETIME.

Default value is "d"

Definition

```
Public Property HotSpotDateTimeFormat() As String
```

Parameters

String Custom datetime string format (see Visual Studio .NET help for a list of supported codes)

Returns

Current setting.

HotSpotsNumericFormat

Description

Set or returns datetime format for tooltips when a numeric value is shown.

Default value is "N" (which includes 2 decimals).

Definition

```
Public Property HotSpotNumericFormat() As String
```

Parameters

String	Custom numeric string format (see Visual Studio .NET help for a list of supported codes) For example, use "0" (zero) for integer numbers only.
--------	---

Returns

Current setting.

Image

Description

Returns last chart rendering as an bitmap object.

Definition

```
ReadOnly Property Image() As Bitmap
```

Parameters

None.

Returns

Last chart rendered.

ItemsHighlight

Description

Use this property to explode any number of sectors in a pie or doughnut chart (either 2d or 3d style).

Definition

`Property ItemsHighlight() As String()`

Parameters

`String()` Array of strings. Each item corresponds to a sector of the pie. If the item contains any character, the sector is exploded.

For example:

```
Dim explode() as string = String() { "1", "", "OK", "This", "" }
```

The previous array would result in exploding the first, third and fourth sectors. "1", "OK" and "This" are irrelevant in this version. What is important is that the string item must be different from "". Future versions of the library may use the string item to show a particular message over each exploded sector.

Returns

Current explode configuration as an array of Strings.

ItemsLegend

Description

Array of strings containing item names for each value in a serie (all series share the same item names). This is useful in pie charts, where each value may have a description or name associated.

Definition

```
Property ItemsLegend() As String()
```

Parameters

String() Array of strings. It is important to note that the length of this array must be equal to the number of values in any serie (all series should have the same number of values).

Returns

Current explode configuration as an array of Strings.

LegendAlignment

Description

Specify location for legend.

Default value is `STYLEALIGNMENT_MIDDLE_RIGHT`

Definition

`Property` `LegendAlignment()` `As` `STYLEALIGNMENT`

Parameters

`STYLEALIGNMENT` Location enumeration. Can be one of the following:

```
STYLEALIGNMENT_MIDDLE_RIGHT = 0
STYLEALIGNMENT_BOTTOM_RIGHT = 1
STYLEALIGNMENT_BOTTOM_CENTER = 2
STYLEALIGNMENT_BOTTOM_LEFT = 3
STYLEALIGNMENT_MIDDLE_LEFT = 4
STYLEALIGNMENT_TOP_LEFT = 5
STYLEALIGNMENT_TOP_CENTER = 6
STYLEALIGNMENT_TOP_RIGHT = 7
```

Note that top center and bottom center options will draw the legend horizontally while the rest of options will draw the legend vertically.

Returns

Current explode configuration as an array of Strings.

LegendBackColor

Description

Specify color for legend's background. Only applies to the legend title.

Default value is `Color.White`

Definition

```
Public Property LegendBackColor() As Color
```

Parameters

Color Any color may be used (you can create any color using `Color.FromArgb` static function).

Returns

Current legend's backcolor.

LegendColor

Description

Set legend text color.

Default value is `Color.Black`

Definition

`Property LegendColor() As Color`

Parameters

Color Any color may be used (you can create any color using `Color.FromArgb` static function).

Returns

Current color of the legend text.

LegendFont

Description

Sets legend's text font.

Default value is `New Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point).`

Definition

```
Property LegendFont () As Font
```

Parameters

Font	Any font object may be used.
------	------------------------------

Returns

Current legend's text font.

LegendText

Description

Set legend's title.

Default value is "Legend"

Definition

```
Property LegendText() As String
```

Parameters

String Legend's title.

Returns

Current legend's title.

LicenseeIdentifier

Description

Set serial number and unlock the trial version. This function should be the first call to the control just after its creation.

Definition

```
Property LicenseeIdentifier() As String
```

Parameters

String Unique licensee's serial number.

Returns

Current serial number.

MarginBottom

Description

Specify the inner bottom margin in pixels (distance between the border and the lower limit of the chart).

Default value is 20

Definition

```
Property MarginBottom() As Integer
```

Parameters

Integer	Distance in pixels
---------	--------------------

Returns

Current preference.

MarginLeft

Description

Specify the inner left margin in pixels (distance between the border and the left limit of the chart).

Default value is 20

Definition

```
Property MarginLeft() As Integer
```

Parameters

Integer	Distance in pixels
---------	--------------------

Returns

Current preference.

MarginRight

Description

Specify the inner right margin in pixels (distance between the border and the right limit of the chart).

Default value is 20

Definition

```
Property MarginRight() As Integer
```

Parameters

Integer	Distance in pixels
---------	--------------------

Returns

Current preference.

MarginTop

Description

Specify the inner top margin in pixels (distance between the border and the top limit of the chart).

Default value is 20

Definition

```
Property MarginTop() As Integer
```

Parameters

Integer	Distance in pixels
---------	--------------------

Returns

Current preference.

MarkHighest

Description

Whether maximum value of the entire chart should be highlighted from the rest.

Default value is `False`

Definition

```
Property MarkHighest() As Boolean
```

Parameters

Boolean	Highlight or not the maximum value of the chart.
---------	--

Returns

Current preference.

MarkLowest

Description

Whether minimum value of the entire chart should be highlighted from the rest.

Default value is `False`

Definition

```
Property MarkLowest () As Boolean
```

Parameters

Boolean	Highlight or not the minimum value of the chart.
---------	--

Returns

Current preference.

PieBorderStyle

Description

Specify the line style for the borders of pie segments (2D and 3D pies).

Default value is `Pen(Color.Transparent)` . If this property is not set, then borders will share the segment color.

Definition

```
Property PieBorderStyle As Pen = new Pen(Color.Transparent)
```

Parameters

Pen object	Pen object with line style, width, color...
------------	---

Returns

Current setting value.

PointSize

Description

Specify the size in pixels of points (see `PointStyle` or `ShowPoints` properties).

Default value is 4

Definition

```
Property PointSize() As Integer
```

Parameters

Integer	Point size in pixels
---------	----------------------

Returns

Current points' size.

PointStyle

Description

Specify the point style when a point is drawn over the chart. Applies to point or scatter charts, and line or area charts when ShowPoints is enabled.

Default value is `STYLEPOINT_CIRCLE`

Definition

`Property` `PointStyle()` `As` `STYLEPOINT`

Parameters

POINTSTYLE	Can be one of the following values: <code>STYLEPOINT_CIRCLE = 0</code> <code>STYLEPOINT_RECTANGLE = 1</code> <code>STYLEPOINT_TRIANGLE = 2</code>
------------	--

Returns

Current preference.

Quality

Description

Set the overall rendering quality factor.

Default value is `SmoothingMode.HighQuality`

Definition

```
Public Property Quality() As SmoothingMode
```

Parameters

SmoothingMode

Choose one of the following values:

HighSpeed	(Faster)
Default	(Standard)
AntiAlias	(Better but slower)
HighQuality	(Best, slowest)

Returns

Current quality factor.

Reset

Description

Clears all data and assign default values to all properties.
Generally, Reset is used before any other library commands.

Definition

```
Public Sub Reset()
```

Parameters

None.

Returns

None..

Resolution

Description

Specify the output resolution in DPI (dots per inch).

Default value is 72 (screen resolution). Increase this value for printing purposes (for example 200 or 300).

Definition

```
Property Resolution() As Integer
```

Parameters

Integer	Maximum value recommended depends on memory available.
---------	--

Returns

Current resolution.

Series

Description

Assign values to the chart.

Values in the chart are expressed as an array of ArrayList. The array has one arraylist for each serie. The arraylist (each serie) may have any number of values. A value in an arraylist usually is an integer/single/double value, but it may be also an array in case of a multidimensional chart, like scatter (2 dimensions), bubble and high-low-close (3 dimensions) or candlestick (4 dimensions).

For example, in a 2d bar chart, each bar only needs a single value (1 dimension). In scatter charts, each value must be a pair of single (x and y position for point) expressed as an array of single. In bubble charts, each value must be an array of three singles (x, y and size of the bubble), an so on.

Instead of using this property, you may also use the SeriesFactory helper class. See "Populating the chart with values" chapter for more information.

Definition

```
Property Series() As ArrayList()
```

Parameters

ArrayList()	Array of ArrayLists
-------------	---------------------

Returns

Current chart values.

SeriesColor

Description

One dimension array of colors for series. Each color in the array corresponds to a serie (all values in a serie share the same colors, except for pie and doughnut charts, where each value in array corresponds to one value).

Definition

```
Property SeriesColor() As Color()
```

Parameters

Color() Array of System.Drawing.Color values.

Returns

Current colors for series.

SeriesBorderStyle

Description

One dimension array of pen objects for each series' bar borders. Each pen in the array corresponds to a serie (all values in a serie share the same pen style).

Definition

```
Property SeriesBorderStyle () As Pen()
```

Parameters

Pen ()	Array of System.Drawing.Pen objects
--------	-------------------------------------

Returns

Current pen styles for series.

SeriesFillStyle

Description

One dimension array of brush objects for series. Each brush in the array corresponds to a serie (all values in a serie share the same brush style). This property only applies to Line Chart.

Definition

```
Property SeriesFillStyle() As Brush()
```

Parameters

Brush ()	Array of System.Drawing.Brush object
----------	--------------------------------------

Returns

Current brush styles for series.

SeriesLegend

Description

One dimension array of string values containing series titles.

Definition

```
Property SeriesLegend() As String()
```

Parameters

String()	Array of series titles.
----------	-------------------------

Returns

Current series titles.

SeriesLineStyle

Description

Array of System.Drawing.Pen objects containing a pen object for each serie. This property only applies to Line Chart style.

Definition

```
Property SeriesLineStyle() As Pen()
```

Parameters

String()	Array of Pen objects
----------	----------------------

Returns

Current pen objects used for line drawing.

SeriesPicture

Description

Array of Image objects containing a Image object for each serie. This property only applies to Picture Chart style.

Definition

```
Property SeriesPictures() As Image()
```

Parameters

Image()	Array of Image objects
---------	------------------------

Returns

Current image objects used for picture drawings.

SettingsFromString

Description

This function will apply look and feel settings packed in a string value (the string may be large – from 4 to 50+ Kbytes, depending on which settings saved, pictures, ...)

Definition

```
Public WriteOnly Property SettingsFromString() As String
```

Parameters

String	Look and feel settings packed in a base64 encoded string
--------	--

Returns

Nothing

ShadowColored

Description

Use colored shadows. Set this property to false and shadows will be gray.

Default value is false.

Definition

`Property ShadowColored() As Boolean`

Parameters

Boolean	True=colored shadows, False=gray shadows.
---------	---

Returns

Current preference.

ShadowOffset

Description

Distance of the shadow in pixels. Note that shadows will begin to dim if you set this value too high (this is a nice effect).

Default value is 5.

Definition

```
Property ShadowOffset () As Integer
```

Parameters

Integer	Distance in pixels.
---------	---------------------

Returns

Current preference.

ShadowSmooth

Description

Use diffuse/soft shadows. If set to false shadows will be solid.

Default value is true.

Definition

`Property ShadowColored() As Boolean`

Parameters

Boolean	True=soft shadows, False=solid shadows.
---------	---

Returns

Current preference.

ShowDataTable

Description

Whether a table with series' data is shown at the bottom of the control.

Default value is true.

Definition

`Property ShowDataTable() As Boolean`

Parameters

Boolean	Show or not the datatable
---------	---------------------------

Returns

Current preference.

ShowDividerX

Description

Whether vertical separator lines are drawn over the chart, grouping the items of different series. Also affects to vertical lines in the datatable.

Default value is true.

Definition

```
Property ShowDividerX() As Boolean
```

Parameters

Boolean	Show or not the vertical lines
---------	--------------------------------

Returns

Current preference.

ShowItemsLegend

Description

Useful for pie or doughnut charts, where items usually have names. Set this property to TRUE to show the items names on the datatable. Must be used in conjunction with ItemsLegend property.

Default value is false

Definition

`Property ShowItemsLegend() As Boolean`

Parameters

Boolean	Show or not the names of the items on the datatable.
---------	--

Returns

Current preference.

ShowLegend

Description

Show or hide the legend. To specify the position for the legend use LegendAlignment property.

Default value is true.

Definition

`Property ShowLegend() As Boolean`

Parameters

Boolean	Show or not the legend
---------	------------------------

Returns

Current preference.

ShowLegendInDataTable

Description

Show or hide the legend names on the first column of the datatable.

Default value is true.

Definition

```
Public Property ShowLegendInDataTable() As Boolean
```

Parameters

Boolean	Show or not the legend in the datatable
---------	---

Returns

Current preference.

ShowPoints

Description

Draw additional markers over some styles of charts, like line or area charts.

Default value is false.

Definition

`Property ShowPoints() As Boolean`

Parameters

Boolean	Show or not points or additional markers.
---------	---

Returns

Current preference.

ShowSolidColors

Description

Deactivate gradients and use only plain colors. When you specify a color for an element (say a bar serie), when ShowSolidColors = false, the bar is drawn using a gradient of colors based upon the color you specified for the element. The range for the gradient is calculated adding a 20% of dark to the original color for one limit and adding a 20% of brightness to the original color for the other limit. Setting this property to TRUE, will ignore the gradient and use the same color for the entire element.

Default value is false.

Definition

`Property ShowSolidColors() As Boolean`

Parameters

Boolean	True will use only plain colors, False will use gradients.
---------	--

Returns

Current preference.

ShowSurface3d

Description

Draws a ocean-like surface for 3d charts. Negatives values sinks into the water while positive values arise. This property is useful to enhance the positive values versus the negative values.

Default value is false.

Definition

`Property ShowSurface3d() As Boolean`

Parameters

Boolean	Show or not "surface line".
---------	-----------------------------

Returns

Current preference.

ShowValues

Description

Show or not the values over the chart.

Default value is true.

Definition

`Property ShowValues() As Boolean`

Parameters

Boolean	Show or not the values over the chart.
---------	--

Returns

Current preference.

ShowValuesMax

Description

Maximum value for items' captions to be displayed.

Default value is Double.Max (unlimited).

Definition

`Property ShowValuesMax() As Double`

Parameters

Double	Max range.
--------	------------

Returns

Current preference.

ShowValuesMin

Description

Minimum value for items' captions to be displayed.

Default value is Double.Min (unlimited).

Definition

`Property ShowValuesMin() As Double`

Parameters

Double	Min range.
--------	------------

Returns

Current preference.

ShowXAxis

Description

Show or not the X-Axis (and its values).

Default value is False.

Definition

`Property ShowXAxis() As Boolean`

Parameters

Boolean	Show or not the X-axis (ant is values).
---------	---

Returns

Current preference.

SlantX

Description

Specify the amount in pixels for X displacement resulting in a pseudo Y-Axis rotation. Only applies to 3d charts.

Default value is 30.

Definition

```
Public Property SlantX() As Single
```

Parameters

Single	Amount in pixels for the X-axis displacement.
--------	---

Returns

Current preference.

SlantY

Description

Specify the amount in pixels for Y displacement resulting in a pseudo X-Axis rotation. Only applies to 3d charts.

Default value is 30.

Definition

```
Public Property SlantX() As Single
```

Parameters

Single	Amount in pixels for the X-axis displacement.
--------	---

Returns

Current preference.

Style

Description

Set the chart style (type). Default value is `STYLE2D3D.STYLE2D_BAR`.

Definition

Property `Style()` **As** `STYLE2D3D`

Parameters

`STYLE2D3D`

Can be one of the following values:

```
STYLE2D_BAR = 1
STYLE2D_ACUM = 2
STYLE2D_LINE = 3
STYLE2D_POINT = 4
STYLE2D_PIE = 5
STYLE2D_STACKED_AREA = 6
STYLE2D_CANDLESTICK = 7
STYLE2D_HIGHLOWCLOSE = 8
STYLE2D_HIGHLOW = 9
STYLE2D_SPLINE = 10
STYLE2D_XY = 11
STYLE2D_BUBBLE = 12
STYLE2D_PYRAMID = 13
STYLE2D_PICTURES = 14
STYLE2D_DOUGHNUT = 15
STYLE2D_CYLINDER = 16
STYLE2D_ROUNDED_BAR = 17
STYLE2D_HORIZONTAL_BAR = 18
STYLE2D_ACUM_FIXED = 19
STYLE2D_XY_DATETIME = 20
STYLE2D_GAUGE = 22
STYLE2D_LINE_DATETIME = 21
STYLE2D_XY_SPLINE = 23
STYLE2D_XY_SPLINE_DEV = 24
STYLE2D_GANTT = 25
STYLE2D_RADAR = 26
STYLE3D_ACUM_BAR = 101
STYLE3D_LINE = 102
STYLE3D_STACKED_AREA = 103
STYLE3D_COLUMN = 104
STYLE3D_PIE = 105
STYLE3D_DOUGHNUT = 106
STYLE3D_CYLINDER = 107
STYLE3D_SPLINE = 108
STYLE3D_SPLINE_AREA = 109
STYLE3D_SPLINE_STACKED_AREA = 110
STYLE3D_STACKED_BAR = 111
STYLE3D_STACKED_COLUMN = 112
```

Returns

Current preference.

Thickness

Description

Specify the width in pixels of cylinders in a cylinder chart.

Default value is 30 pixels.

Definition

```
Property Thickness() As Integer
```

Parameters

Integer	Amount in pixels for the width of cylinder charts.
---------	--

Returns

Current preference

Title

Description

Specify the main title of the chart. Set to "" to hide the title.

Default value is "Title".

Definition

```
Property Title() As String
```

Parameters

String	Chart's title
--------	---------------

Returns

Current chart's title.

TitleColor

Description

Specify the text color for main title.

Default value is black.

Definition

```
Property TitleColor() As Color
```

Parameters

Color	Text color for main title.
-------	----------------------------

Returns

Current text color of main title.

TitleFont

Description

Specify the font for the main title of the chart.

Default value: `Font("Arial", 15, FontStyle.Bold, GraphicsUnit.Point)`

Definition

`Property TitleFont() As Font`

Parameters

Font	Font of chart's title
------	-----------------------

Returns

Current font of chart's title.

ValuesAlignment

Description

Specify the alignment for values in the datatable.

Default value is `StringAlignment.Far`.

Definition

`Property` ValuesAlignment() `As` StringAlignment

Parameters

StringAlignment	Far (right), Center or Near (left)
-----------------	------------------------------------

Returns

Current setting.

ValuesBackColor

Description

Specify the text bgcolor for values. To show/hide vales use ShowValues property.

Default value is `Color.FromArgb(255, 192, 255, 255)`.

Definition

`Property ValuesBackColor() As Color`

Parameters

Color	Text color for values.
-------	------------------------

Returns

Current background color of values.

ValuesColor

Description

Specify the text color for values. To show/hide vales use ShowValues property.

Default value is black.

Definition

```
Property ValuesColor() As Color
```

Parameters

Color	Text color for values.
-------	------------------------

Returns

Current text color of values.

ValuesFont

Description

Specify the font used for values over the chart (for datatable see DataTableFont property).

Default value is `Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point)`

Definition

```
Property ValuesFont() As Font
```

Parameters

Font	Any font object.
------	------------------

Returns

Current setting.

ValuesFormat

Description

Specify the string format for values. Sample format are "p" (percentage) and "N" (standard number). Refer to .NET documentation for standard and custom string formats.

Default value is "N".

Definition

`Property ValuesFormat() As String`

Parameters

String	Format string for values.
--------	---------------------------

Returns

Current format string.

ValuesLocationStyle

Description

Specify the location style for values.

Default value is `STYLECAPTIONLOCATION_SMART_PLACEMENT`.

Definition

`Property ValuesLocationStyle()` `As` `STYLECAPTIONLOCATION`

Parameters

<code>STYLECAPTIONLOCATION</code>	Location style. Can be one of the following values: <code>STYLECAPTIONLOCATION_DEFAULT = 0</code> <code>STYLECAPTIONLOCATION_INSIDE = 1</code> <code>STYLECAPTIONLOCATION_SMART_PLACEMENT = 2</code>
-----------------------------------	---

Note: Inside only applies to 2d bar charts. Smart placement will avoid label overlapping. We suggest you experiment with different values.

Returns

Current location style for values.

ValuesStyle

Description

Specify the appearance of values.

Default value is `STYLECAPTION_ROUNDED_SQUARE`.

Definition

`Property` `ValuesStyle()` `As` `STYLECAPTION`

Parameters

<code>STYLECAPTION</code>	Appearance style for values. Can be one of the following values: <code>STYLECAPTION_STANDARD = 0</code> <code>STYLECAPTION_ROUNDED_SQUARE = 1</code>
---------------------------	--

Returns

Current values style.

WallsBackColorPrincipal

Description

Specify the main color for grid background.

Default value is `Color.FromArgb(64, 148, 232, 255)`.

Definition

`Property WallsBackColorPrincipal() As Color`

Parameters

`Color` Any `System.Drawing.Color`

Returns

Current preference.

WallsBackColorSecondary

Description

Specify the secondary color for grid background.

Default value is `Color.FromArgb(128, 148, 232, 255)`.

Definition

`Property WallsBackColorSecondary() As Color`

Parameters

`Color` Any `System.Drawing.Color`

Returns

Current preference.

WallsBackStyle

Description

Specify the fill style for walls in 3d charts.

Default value is STYLEWALLBACKGROUND_BANDS.

Definition

`Property WallsBackStyle() As STYLEWALLBACKGROUND`

Parameters

STYLEWALLBACKGROUND

Can be one of the following values:

```
STYLEWALLBACKGROUND_TRANSPARENT = 0
STYLEWALLBACKGROUND_BANDS = 1
STYLEWALLBACKGROUND_CHESS = 2
```

Returns

Current preference.

WallsStyle

Description

Specify the style of grids in 3d charts.

Default value is STYLEWALLS_NOWALLS.

Definition

```
Public Property WallStyle() As STYLEWALLS
```

Parameters

```
STYLEWALLS Can be one of the following values:  
    STYLEWALLS_NOWALLS = 0  
    STYLEWALLS_BOXED = 1  
    STYLEWALLS_VISIBLE = 2
```

Returns

Current preference.

XAxisColor

Description

Specify the color for X-axis items.

Default value is Color.Black.

Definition

```
Property XAxisColor() As Color
```

Parameters

Color Any System.Drawing.Color

Returns

Current preference.

XAxisDividersCount

Description

Number of X-axis line dividers and labels.

Default value is 0 (automatic).

Definition

```
Property XAxisDividersCount() As Long
```

Parameters

Long Number of X-Axis dividers.

Returns

Current setting.

XAxisHeight

Description

Height of the space allocated for X-Axis labels. Maximum effective value will be the size needed to show any X-Axis label without word-wrap. If your labels are too long, try reducing this value so word-wrapping may occur.

Default value is 1000.

Definition

```
Property YAxisHeight () As Integer
```

Parameters

Integer Height in pixels for X-Axis labels space

Returns

Current height.

XAxisLabelsRotated

Description

Labels of XAxis will be rotated 45 degrees if this property is set to True. Rotating labels reduces the space they occupy and improve results when labels are too long.

Default value is False.

Definition

`Property` `XAxisLabelsRotated()` `As Boolean`

Parameters

`Boolean` Rotate or not the X-Axis labels.

Returns

Current preference.

XAxisLabelsRotated90

Description

Labels of XAxis will be rotated 90 degrees if this property is set to True. Rotating labels reduces the space they occupy and improve results when labels are too long.

Default value is False.

Definition

`Property` `XAxisLabelsRotated90()` `As Boolean`

Parameters

`Boolean` Rotate or not the X-Axis labels.

Returns

Current preference.

XAxisNumericFormat

Description

Specify the string format for X-axis values. Sample format are "p" (percentage) and "N" (standard number). Refer to .NET documentation for standard and custom string formats. This property should be used in scatter charts.

Default value is "N".

Definition

```
Property xAxisNumericFormat () As String
```

Parameters

String	Format string for values.
--------	---------------------------

Returns

Current format string.

XAxisScaleMaximum

Description

Set the upper limit for the X-Axis scale. Must be used with `XAxisScaleMode = SCALEMODE_FIXED`.

Default value is 0.

Definition

`Property` `XAxisScaleMaximum()` `As Single`

Parameters

`Single` Upper limit for the X-Axis scale.

Returns

Current preference.

XAxisScaleMaximumDateTime

Description

Set the upper limit for the X-axis scale where XAxisScaleMode is set to SCALEMODE_FIXED_DATETIME.

Default value is 0.

Definition

`Property XAxisScaleMaximumDateTime() As DateTime`

Parameters

DateTime	Upper limit for the X-Axis scale.
----------	-----------------------------------

Returns

Current preference.

XAxisScaleMinimum

Description

Set the lower limit for the X-Axis scale. Must be used with `XAxisScaleMode = SCALEMODE_FIXED`.

Default value is 0.

Definition

`Property` `XAxisScaleMinimum()` `As Single`

Parameters

`Single` Lower limit for the X-Axis scale.

Returns

Current preference.

XAxisScaleMinimumDateTime

Description

Set the lower limit for the X-axis scale where XAxisScaleMode is set to SCALEMODE_FIXED_DATETIME.

Default value is 0.

Definition

`Property` XAxisScaleMinimumDateTime() `As` DateTime

Parameters

DateTime	Lower limit for the X-Axis scale.
----------	-----------------------------------

Returns

Current preference.

XAxisScaleMode

Description

Specify the scale mode for X-Axis rule. By default, X-Axis rule values are calculated automatically from the values of the chart (minimum value of the X-Axis rule equals the minimum X of any value in the chart). Set this property to SCALEMODE_FIXED so you can set the minimum and maximum values for the rule (see XAxisScaleMinimum and XAxisScaleMaximum properties).

Default value is SCALEMODE.AUTOMATIC

Definition

```
Property XAxisScaleMode () As SCALEMODE
```

Parameters

SCALEMODE

Can be one of the following values:

```
SCALEMODE_AUTOMATIC = 0  
SCALEMODE_FIXED = 1  
SCALEMODE_FIXED_DATETIME = 2  
SCALEMODE_AUTOMATIC_MINIMUM_ZERO = 3
```

Returns

Current preference.

XAxisTitle

Description

Set the title for the X-Axis. Set this property to "" to hide the X-Axis title.

Default value is "" (title hidden).

Definition

```
Property XAxisTitle() As String
```

Parameters

String Title for X-axis.

Returns

Current X-axis' title.

XAxisTitleFont

Description

Specify the font to be used for the X-Axis title.

Default value is `Font("Arial", 11, FontStyle.Bold, GraphicsUnit.Point)`.

Definition

```
Property XAxisTitleFont() As Font
```

Parameters

Font Any System.Drawing.Font object.

Returns

Current X-axis' title font.

XAxisValuesFont

Description

Specify the font to be used for the X-Axis labels.

Default value is `Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point)`.

Definition

`Property` `XAxisValuesFont()` `As` `Font`

Parameters

`Font` Any `System.Drawing.Font` object.

Returns

Current X-axis' title font.

YAxisColor

Description

Specify the color for Y-axis items.

Default value is Color.Black.

Definition

```
Property YAxisColor() As Color
```

Parameters

Color Any System.Drawing.Color

Returns

Current preference.

YAxisDividersCount

Description

Number of Y-Axis line dividers. To hide Y-Axis dividers set this property to 0 (zero). Set this property to 10 and each line would be useful to mark intervals of 10%.

Default value is 10.

Definition

`Property YAxisDividersCount () As Long`

Parameters

`Long` Number of Y-Axis dividers.

Returns

Current preference.

YAxisIndicators

Description

Array or Y_AXIS_INDICATORS structure objects. Each indicator represents an horizontal line over the chart useful for marking special limits.

Default value is Nothing.

Y_AXIS_INDICATORS structure:

```
Public Structure YAXIS_INDICATORS
    Dim Caption As String           \ Text to show
    Dim CaptionColor As Color       \ Text-color
    Dim CaptionBackColor As Color  \ Text-backcolor
    Dim CaptionFont As Font        \ Text-font
    Dim CaptionStyle As STYLECAPTION \ Text-backstyle (standard or rounded filled)
    Dim Value As Single            \ Value for the Y-axis location of the line
    Dim LineStyle As Pen           \ Pen object for drawing the line
End Structure
```

Definition

```
Public Property YAxisIndicators() As YAXIS_INDICATORS()
```

Parameters

Long Number of Y-Axis dividers.

Returns

Current preference.

YAxisLabelsRotated

Description

Labels of Y-Axis will be rotated 45 degrees if this property is set to True. Rotating labels reduces the space they occupy and improve results when labels are too long.

Default value is False.

Definition

`Property YAxisLabelsRotated() As Boolean`

Parameters

`Boolean` Rotate or not the Y-Axis labels.

Returns

Current preference.

YAxisLocation

Description

You may specify if Y-Axis is drawn on the left or on the right side of the chart.

Default value is `STYLEYAXISLOCATION_LEFT`.

Definition

`Property` YAxisLocation() `As` STYLEYAXISLOCATION

Parameters

STYLEYAXISLOCATION	May be one of the following values:
	STYLEYAXISLOCATION_LEFT = 0
	STYLEYAXISLOCATION_RIGHT = 1

Returns

Current preference.

YAxisNumericFormat

Description

Specify the string format for Y-axis values. Sample format are "p" (percentage) and "N" (standard number). Refer to .NET documentation for standard and custom string formats.

If you want big numbers to be divided, use the following formats:

"0," -> one comma will divide the value by 1000.

"0,," -> two commas will divide the value by 1000000.

"0,.00" -> one comma plus a dot and 00 will show the value divided by 1000 and two decimals.

Default value is "N".

Definition

`Property YAxisNumericFormat () As String`

Parameters

String	Format string for values.
--------	---------------------------

Returns

Current format string.

YAxisScaleMaximum

Description

Set the upper limit for the Y-Axis scale. Must be used with YAxisScaleMode = SCALEMODE_FIXED.

Default value is 0.

Definition

`Property YAxisScaleMaximum() As Single`

Parameters

Single Upper limit for the Y-Axis scale..

Returns

Current preference.

YAxisScaleMinimum

Description

Set the lower limit for the Y-Axis scale. Must be used with YAxisScaleMode = SCALEMODE_FIXED.

Default value is 0.

Definition

`Property YAxisScaleMinimum() As Single`

Parameters

Single Lower limit for the Y-Axis scale..

Returns

Current preference.

YAxisScaleMode

Description

Specify the scale mode for Y-Axis rule. By default, Y-Axis rule values are calculated automatically from the values of the chart (minimum value of the Y-Axis rule equals the minimum Y of any value in the chart). Set this property to SCALEMODE_FIXED so you can set the minimum and maximum values for the rule (see YAxisScaleMinimum and YAxisScaleMaximum properties).

Default value is SCALEMODE.AUTOMATIC

Definition

`Property YAxisScaleMode () As SCALEMODE`

Parameters

SCALEMODE Can be one of the following values:
SCALEMODE_AUTOMATIC = 0
SCALEMODE_FIXED = 1
SCALEMODE_AUTOMATIC_MINIMUM_ZERO = 3

Returns

Current preference.

YAxisShowDividers

Description

Show or not the Y-axis line dividers (note, this property only controls line drawing – see YAxisDividerCount to control the number of Y-Axis labels).

Default value is True.

Definition

`Property YAxisShowDividers() As Boolean`

Parameters

Boolean	Show/hide y-axis line dividers.
---------	---------------------------------

Returns

Current setting.

YAxisTitle

Description

Set the title for the Y-Axis. Set this property to "" to hide the Y-Axis title.

Default value is "" (title hidden).

Definition

```
Property YAxisTitle() As String
```

Parameters

String Title for Y-axis.

Returns

Current Y-axis' title.

YAxisTitleFont

Description

Specify the font to be used for the Y-Axis title.

Default value is `Font("Arial", 11, FontStyle.Bold, GraphicsUnit.Point)`.

Definition

`Property YAxisTitleFont() As Font`

Parameters

Font Any `System.Drawing.Font` object.

Returns

Current Y-axis' title font.

YAxisValuesFont

Description

Specify the font to be used for the Y-Axis labels.

Default value is `Font("Arial", 12, FontStyle.Regular, GraphicsUnit.Point)`.

Definition

`Property YAxisValuesFont() As Font`

Parameters

Font Any System.Drawing.Font object.

Returns

Current Y-axis' title font.