Graph Control

See Also Properties Constants

The graph control allows you to design graphs interactively on your forms. At run time, you can send new data to the graphs and draw them, print them, copy them onto the Clipboard, or change their styles and shapes.

File Name

GRAPH32.OCX

Class Name

GraphicsServer

Remarks

The graph control acts as a link between your application and the Graphics Server graphing and charting library.

At design time, the graph control has an automatic redraw capability. Every time you change a property, the control redraws the graph so that you can see the effects of the change. You can enter data for the graph either at design time or at run time. At run time, when graph is given new data and style options, it combines these new values with your design-time values.

As a design aid, the graph control automatically generates random data at design time to give you an idea of what your graph will look like.

Distribution Note When you create and distribute applications that use the graph control, you should install the appropriate files in the customer's WINDOWS \SYSTEM subdirectory.

See Also

Property Types and Arrays
Graph Types and Negative Values
Graph Control Constants
Required Files
Graph Control Extended Version

Property Types and Arrays, Graph Control

Example

The following table describes array properties for the graph control.

Property	Description
GraphData	Values to be graphed (this is a two-dimensional array when there are multiple data sets).
ColorData	Colors of bars, pie slices, lines, and so on.
ExtraData	Extra style options (for example, which pie slices to explode).
LabelText	Labels.
LegendText	Legends.
PatternData	Pattern and line styles.
SymbolData	Symbols for lines, legends, and so on.
XPosData	X-variable data for scatter graphs.

Array properties are controlled through two simple properties: ThisSet and ThisPoint. ThisSet is the index for the data you entered with the GraphData property. ThisPoint references the individual data points for the set specified by the ThisSet property. Both have a minimum value of 1.

For example, if you set ThisSet to 1, ThisPoint to 5, and LabelText to "Friday," the fifth label of the first data set is set to the text string "Friday."

The AutoInc property, when set to 1 (on), automatically increments ThisPoint and ThisSet every time you enter an array property value.

At run time, when you dynamically create a new instance of a control array, you must reassign all data associated with array properties.

The overall dimensions of the arrays are determined by the properties NumSets and NumPoints. ThisSet and ThisPoint cannot exceed NumSets and NumPoints, respectively, and the AutoInc property functions monitor their current values. NumSets and NumPoints also determine what the graphs look like. For example, if you want to graph three data sets, each containing ten points, set NumSets to 3 and NumPoints to 10, and then enter the GraphData values.

DataReset is another property associated with arrays. It allows you to clear all the data held in any or all of the array properties. For example, if you haven't set any LabelText strings, the graph control labels your graph 1, 2, 3, and so on. Deleting all your labels individually would have the effect of displaying no labels (that is, labels exist but they are all null). Using DataReset sets the LabelText strings back to their original numeric values of 1, 2, 3, and so on.

Property Types and Arrays Example, Graph Control

At design time, to enter a data set of five points, set the AutoInc property to 1 (on), select the GraphData property in the Object Inspector, and enter the following five values, pressing ENTER between each number. For example:

10 ENTER

9 ENTER

8 ENTER

7 ENTER

6 ENTER

Other information about graphs, such as labels and legends, can be entered in the same manner.

To change the values of a graph at run time, you write code. The following two examples would cause the same property value changes as in the previous example:

```
{ Example 1 }
GraphicsServer1.AutoInc := 1;
GraphicsServer1.GraphData := 10;
GraphicsServer1.GraphData := 9;
GraphicsServer1.GraphData := 8;
GraphicsServer1.GraphData := 7;
GraphicsServer1.GraphData := 6;
GraphicsServer1.DrawMode := 2;

{ Example 2 }
GraphicsServer1.AutoInc := 1;
for I := 1 to 5 do
   GraphicsServer1.GraphData := 11 - I;
GraphicsServer1.DrawMode := 2;
```

Graph Types and Negative Values, Graph Control

Certain graph types cannot handle negative data meaningfully. They are the following:

- Pie charts (2D & 3D)
- Stacked Bar graphs
- Gantt charts
- Area graphs
- Polar graphs

For these graphs, negative data is forced to a positive number; however, the data is not permanently changed. Changing to a graph type for which negative values are meaningful restores the original data.

Required Files

The files required by the Graph control are:
GRAPH32.OCX
GSW32.EXE
GSWDLL32.DLL

Properties

The properties that apply *only* to the Graph control are:

AutoInc
Background
BottomTitle
ColorData
CtlVersion
DataReset
DrawMode
DrawStyle
ExtraData
FontFamily
FontSize
FontStyle
FontUse
Foreground
GraphCaption
GraphData
GraphStyle
GraphTitle
GraphType
GridStyle
ImageFile
IndexStyle
LabelEvery
Labels
LabelText
LeftTitle
LegendStyle
LegendText
LineStats
NumPoints
NumSets
Palette
PatternData
PatternedLines
Picture
PrintStyle
RandomData
SeeThru
SymbolData
ThickLines
ThisPoint
ThisSet
TickEvery
Ticks
XPosData
YAxisMax
YAxisMin
YAxisPos
YAxisStyle
YAxisTicks

AutoInc Property, Graph Control

Example

Allows the properties specific to arrays to be set without manually incrementing the ThisPoint counter from ThisPoint = 1 to ThisPoint = NumPoints.

When NumSets > 1, AutoInc goes through all the points and sets them consecutively from ThisPoint = 1 to ThisPoint = NumPoints and from ThisSet = 1 to ThisSet = NumSets.

Syntax

[*form*.]*Graph*.**AutoInc**[:= *setting*]

Remarks

The following table lists the AutoInc property settings for the graph control.

Setting	Description
0	Off
1	(Default) On

When AutoInc is set to a new value (0 or 1), ThisPoint and ThisSet are both reinitialized to 1.

If you set the AutoInc property to 1 (on), when you switch from setting one of the array properties to setting a different one, both ThisPoint and ThisSet are reinitialized to 1.

AutoInc only changes ThisPoint and ThisSet when you set data values. When you get or use data values, ThisPoint and ThisSet are unaffected.

The AutoInc property works for all the properties specific to arrays:

- ColorData
- ExtraData
- GraphData
- LabelText
- LegendText
- PatternData
- SymbolData
- XPosData

Data Type

Integer

AutoInc Example, Graph Control

```
var
    I, J: integer;
begin
   GraphicsServer1.ThisSet := 1;
   for I := 1 to GraphicsServer1.NumSets do
   begin
      GraphicsServer1.ThisPoint := 1;
      for J := 1 to GraphicsServer1.NumPoints do
      begin
         GraphicsServer1.GraphData := J * I;
         if  GraphicsServer1.ThisPoint < GraphicsServer1.NumPoints then
            GraphicsServer1.ThisPoint := GraphicsServer1.ThisPoint + 1;
      end;
      if  GraphicsServer1.ThisSet < GraphicsServer1.NumSets then
         GraphicsServer1.ThisSet := GraphicsServer1.ThisSet + 1;
   end;
   GraphicsServer1.DrawMode := 2;
end;
```

Using the AutoInc property, the preceding code may be rewritten as:

```
GraphicsServer1.AutoInc := 1;
for I := 1 to (GraphicsServer1.NumSets * GraphicsServer1.NumPoints) do
   GraphicsServer1.GraphData := GraphicsServer1.ThisPoint *
GraphicsServer1.ThisSet;
GraphicsServer1.DrawMode := 2;
```

It is not possible to use ThisPoint or ThisSet as counters in **For** loops.

Background Property, Graph Control

Selects the background color of the graph.

Syntax

[*form*.]*Graph*.**Background**[:= *color*]

Remarks

The following table lists the Background property settings for the graph control.

Setting	Description
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light gray
8	Dark gray
9	Light blue
10	Light green
11	Light cyan
12	Light red
13	Light magenta
14	Yellow
15	(Default) White

When you change the background color, the colors for the components of the graph are automatically selected. However, you may change the Foreground and the ColorData properties.

Data Type

Integer (Enumerated)

BottomTitle Property, Graph Control

Example

Places the text string that you provide at the bottom of the graph, parallel to the horizontal axis.

Syntax
[*form*.]*Graph*.**BottomTitle**[:= *string*]

Remarks
This property is ignored for Pie charts.

Data Type
String

BottomTitle Example, Graph Control

The following code places the title, "Title," at the bottom of a graph (GraphicsServer1) when you click a command button and no title currently exists. If the BottomTitle property does have a value, when you click the command button, the title will become blank. To try this example, paste the code into the Declarations section of a form that contains a command button and a graph.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
   GraphicsServer1.RandomData := 1;
   if  GraphicsServer1.BottomTitle := '' then
      GraphicsServer1.BottomTitle := 'Title'
   else
      GraphicsServer1.BottomTitle := '';
   GraphicsServer1.DrawMode := 2;
end;
```

ColorData Property, Graph Control

Selects the colors for each of the data sets on the graph. For pie charts and for bar graphs with NumSets = 1, you should specify a color for each point rather than for each set.

Syntax

[*form*.]*Graph*.**ColorData**[:= *setting*]

Remarks

The following table lists the ColorData property settings for the graph control.

Setting	Description
0	(Default) Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light gray
8	Dark gray
9	Light blue
10	Light green
11	Light cyan
12	Light red
13	Light magenta
14	Yellow
15	White

Once you select one color, colors should be selected for all sets or they are shown in black.

Since this is an array property, the array element is determined by the current value of the ThisPoint property.

When you enter data, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new value, the ThisPoint counter is automatically incremented.

Data Type

Integer (Enumerated)

CtlVersion Property, Graph Control

Gives the current release of your graph control. This property is read-only.

Syntax
[*form*.]*Graph*.**CtlVersion**

Data Type
String

DataReset Property, Graph Control

Allows you to remove any or all of the array information that has been supplied to the graph control.

Syntax

[*form*.]*Graph*.**DataReset**[:= *setting*]

Remarks

The following table lists the DataReset property settings for the graph control.

Setting	Description
0	(Default) None
1	GraphData
2	ColorData
3	ExtraData
4	LabelText
5	LegendText
6	PatternData
7	SymbolData
8	XPosData
9	All Data

The All Data option resets all the data and text arrays.

When you reset an array, you reset it to the original empty state. All properties are set to their default values.

Data Type

Integer (Enumerated)

DrawMode Property, Graph Control

Defines the drawing mode for the graph control.

Syntax

[*form*.]*Graph*.**DrawMode**[:= *mode*]

Remarks

The following table lists the DrawMode property settings for the graph control.

Setting	Description
0	No Action
1	Clear
2	Draw
3	Blit
4	Copy
5	Print
6	Write

DrawMode property values 0 through 3 are recorded when a graph is saved to disk. These values remain the same between design mode and run mode. DrawMode property values 4, 5, and 6 are transient values that trigger the specified actions.

At design time, when you change a property value, the graph is automatically redrawn to show the effect of the change. At run time, the graph is only redrawn when you set DrawMode to 2 (Draw) or 3 (Blit). This allows you to change as many property values as you want before displaying the graph. However, when the form containing a graph is first displayed, the graph is automatically displayed according to the current DrawMode value.

Setting	Action
0	The control is left blank; the graph will not appear. When you want the graph to appear, reset DrawMode to 2.
1	No graph is drawn, but the background of the control is set to the color specified by the Background property. If there is graph caption text, it is displayed in the center of the control.
2	(Default) At design time, this redraws your graph every time you change a property. At run time, resetting DrawMode to 2 causes the graph to be redrawn.
3	There is a brief pause, and then the graph appears all at once. In this mode, the Graphics Server builds a hidden bitmap of the graph and then displays it using the Windows API BitBlit function. This mode is useful if you want to draw a graph, update it with new data, and then instantaneously display the updated graph.
4	The image of the graph is copied onto the Clipboard in either bitmap or metafile format. If DrawMode is set to 3 (Blit), it is in bitmap format; otherwise, it is in metafile format.
5	A high-quality image of the graph can be printed without the form. For more information, see the PrintStyle property.
6	The image of the graph is written to disk as a bitmap (.BMP) or metafile (.WMF). For this option to work, the ImageFile property must be set to provide a name for the file. If DrawMode is set to 3 (Blit), a bitmap is created; otherwise, a metafile is created.

Data Type

Integer (Enumerated)

DrawStyle Property, Graph Control

If the setting is monochrome, this property sets the background to white and all colors to black. If no PatternData, SymbolData, or GraphStyle properties have been set, DrawStyle supplies default patterns and symbols.

Syntax

[*form*.]*Graph*.**DrawStyle**[:= *style*]

Remarks

The following table lists the DrawStyle property settings for the graph control.

Setting	Description
0	Monochrome
1	(Default) Color

Data Type

Integer (Enumerated)

ExtraData Property, Graph Control

Example

The ExtraData property has two purposes:

- To explode pie chart segment(s).
- To specify the color of the sides of a three-dimensional bar chart.

Syntax

[*form*.]*Graph*.**ExtraData**[:= *setting*]

Remarks

The ExtraData property settings for pie charts are listed in the following table.

Setting	Description
0	(Default) Not exploded
1	Exploded

Note With pie charts, when the AutoInc property is set to 1, setting the ExtraData property cycles automatically through the set of pie slices, exploding each slice in turn. To explode a single slice, set AutoInc to 0, set the ThisPoint property to the datapoint you wish to explode, and finally set the ExtraData property to 1.

For three-dimensional bar charts, the ExtraData property settings are described in the following table.

Setting	Description
0	(Default) Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light gray
8	Dark gray
9	Light blue
10	Light green
11	Light cyan
12	Light red
13	Light magenta
14	Yellow
15	White

Since this is an array property, the array element you set is determined by the current value of the ThisPoint property.

When you enter data, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new value, the ThisPoint counter is automatically incremented.

Data Type

Integer (Enumerated)

ExtraData Example, Graph Control

The following code explodes the segments from the center of a three-dimensional pie chart. To try this example, paste the code into the OnCreate event procedure of a form that contains a graph (GraphicsServer1).

```
procedure TForm1.FormCreate(Sender: TObject);
var
   I: integer;
begin
   for I := 1 to 4 do
      GraphicsServer1.GraphData := I;
   ThisPoint := 2 ;
   GraphicsServer1.ExtraData := 1;
   ThisPoint := 4 ;
   GraphicsServer1.ExtraData := 1;
   GraphicsServer1.DrawMode := 2;
   GraphicsServer1.GraphType := 2;
end;
```

FontFamily Property, Graph Control

Selects the font family in which the text specified by the FontUse property is displayed.

Syntax

[*form*.]*Graph*.**FontFamily**[:= *setting*]

Remarks

The following table lists the FontFamily property settings for the graph control.

Setting	Description
0	(Default) Roman
1	Swiss
2	Modern

The graph control specifies font families rather than type faces to avoid having to list all the available fonts, which may vary from one computer to another. A font of the requested generic type (Roman, Swiss, or Modern) is always available, regardless of the Windows configuration used on your computer.

Data Type

Integer (Enumerated)

FontSize Property, Graph Control

Determines the approximate font size in which the text specified by the FontUse property is displayed.

Syntax

[*form*.]*Graph*.**FontSize**[:= *setting*]

Remarks

Enter a value between 50 and 500, inclusive. This value is the percentage of the system font size. The default depends on the setting of the FontUse property.

FontUse setting	FontSize default
0 (graph title)	200%
1 (other titles)	150%
2 (labels)	100%
3 (legend)	100%

FontSize acts as a starting point rather than an absolute setting; the text is reduced, if necessary, to fit into the available space.

Data Type

Integer

FontStyle Property, Graph Control

Determines the style in which the text specified by the FontUse property is displayed.

Syntax

[*form*.]*Graph*.**FontStyle**[:= *setting*]

Remarks

The following table lists the FontStyle property settings for the graph control.

Setting	Description
0	(Default)
1	Italic
2	Bold
3	Bold italic
4	Underlined
5	Underlined italic
6	Underlined bold
7	Underlined bold italic

Data Type

Integer (Enumerated)

FontUse Property, Graph Control

Determines to which text on a graph you will apply the settings for the FontFamily, FontSize, and FontStyle properties.

Syntax

[*form*.]*Graph*.**FontUse**[:= *setting*]

Remarks

The following table lists the FontUse property settings for the graph control.

Setting	Description
0	(Default) Graph title
1	Other titles
2	Labels
3	Legend
4	All text

After you select a text type using FontUse, select the font family, size, and style for that type by setting the FontFamily, FontSize, and FontStyle properties. You can use setting 4 (all text) to make all of your text look alike. For example, you can set all text to display as Swiss family, size 200%, and bold. You can then reuse the FontUse property to change one or more specific text types; for example, you might make all legends bold and underlined.

Note At design time, the values displayed in the Object Inspector for the font family, size, and style are shown for the graph title only.

Data Type

Integer (Enumerated)

Foreground Property, Graph Control

Sets the color of titles, labels, legends, and axes.

Syntax

[*form*.]*Graph*.**Foreground**[:= *setting*]

Remarks

The following table lists the Foreground property settings for the graph control.

Setting	Description
0	Black
1	Blue
2	Green
3	Cyan
4	Red
5	Magenta
6	Brown
7	Light gray
8	Dark gray
9	Light blue
10	Light green
11	Light cyan
12	Light red
13	Light magenta
14	Yellow
15	White
16	(Default) Auto black/white

The graph control automatically uses black or white as its foreground default color. Depending on the background color set, it picks the color that gives the best contrast.

The ColorData property determines the colors of bars, pie slices, and so on.

Data Type

Integer (Enumerated)

GraphCaption Property, Graph Control

Example

Accepts a single line of text that is displayed when DrawMode = 1 (Clear).

Syntax

[*form*.]*Graph*.**GraphCaption**[:= *caption*]

Remarks

The colors of the text and the background can be selected using the Foreground and Background properties.

Data Type

String

GraphCaption Example, Graph Control

The following code displays the text, "Graphics Server," as the caption for GraphicsServer1.

```
GraphicsServer1.GraphCaption := 'Graphics Server';
GraphicsServer1.DrawMode := 1;
```

GraphData Property, Graph Control

Example

Sets the data to be graphed.

Syntax

[*form*.]*Graph*.**GraphData**[:= *data*]

Remarks

Since this is a two-dimensional array property, the array element you set is determined by the current value of the ThisPoint and ThisSet properties.

When you enter data, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new value, the ThisPoint counter is automatically incremented. When it reaches its maximum value (NumPoints), the ThisSet counter is incremented, and ThisPoint is reset to 1. If ThisSet reaches its maximum value (NumSets), it is also reset to 1.

Data Type

Single

GraphData Example, Graph Control

The following code draws the data sets for a bar graph. The data sets are specified by the NumSets property, and the number of points per data set is specified by the NumPoints property. To try this example, paste this code into the OnCreate event procedure of a form that contains a graph (GraphicsServer1).

```
procedure TForm1.FormCreate(Sender: TObject);
var
   I, J: integer;
begin
   GraphicsServer1.ThisSet := 1;
   for I := 1 to GraphicsServer1.NumSets do
   begin
      GraphicsServer1.ThisPoint := 1;
      for J := 1 to GraphicsServer1.NumPoints do
      begin
         GraphicsServer1.GraphData := J*I%;
         if GraphicsServer1.ThisPoint < GraphicsServer1.NumPoints then
            GraphicsServer1.ThisPoint := GraphicsServer1.ThisPoint + 1;
      end;
      if  GraphicsServer1.ThisSet < GraphicsServer1.NumSets then
         GraphicsServer1.ThisSet := GraphicsServer1.ThisSet + 1;
   end;
   GraphicsServer1.DrawMode := 2;
   GraphicsServer1.DrawMode := 4;
end;
```

Using the AutoInc property, the preceding code may be rewritten as:

```
GraphicsServer1.AutoInc := 1;
for I := 1 to (GraphicsServer1.NumSets * GraphicsServer1.NumPoints) do
   GraphicsServer1.GraphData := GraphicsServer1.ThisPoint *
GraphicsServer1.ThisSet;
GraphicsServer1.DrawMode := 2;
GraphicsServer1.DrawMode := 4;
```

GraphStyle Property, Graph Control

Sets the characteristics of each type of graph.

Syntax

[*form*.]*Graph*.**GraphStyle**[:= *type*]

Remarks

The following table describes the GraphStyle property settings for each type of graph.

Graph type	GraphStyle setting		Notes
2D and 3D pie	0	(Default) Lines join labels to pie	If LabelText values are set, then those labels are used; otherwise, the numerical value is used as a label.
	1	No label lines	
	2	Colored labels	
	3	Colored labels without lines	
	4	% Labels	
	5	% Labels without lines	
	6	% Colored labels	
	7	% Colored labels without lines	
2D bar	0	(Default) Vertical bars, clustered if NumSets > 1	If NumSets = 1, then each bar has a different color. If NumSets > 1, then each data set is a different color.
	1	Horizontal	
	2	Stacked	
	3	Horizontal stacked	
	4	Stacked %	
	5	Horizontal stacked%	
3D bar	As preceding, plus:		
	6	Z-clustered	Z-clustered means that the data points for successive sets are drawn in front of the previous one. This gives an illusion of depth.
	7	Horizontal Z-clustered	
Gantt	0	(Default) Adjacent bars	
	1	Spaced bars	Spaced bars have a gap of one bar's width between successive bars.
Line, Log/Lin, and polar	0	(Default) Lines	You can create thick or patterned lines by setting the ThickLine or PatternLine property to 1 (on).
	1	Symbols	
	2	Sticks	
	3	Sticks and symbols	
	4	Lines	
	5	Lines and symbols	

	6	Lines and sticks	
	7	Lines and sticks and symbols	
Area	0	(Default) Stack the data sets	
	1	Absolute	Absolute uses absolute values from Y = 0 (so values can be hidden).
	2	Percentage	Percentage shows the sets as a percentage of the total.
Scatter	0	(Default) Symbols only	Scatter graphs require XPosData to be present.
HLC	0	(Default) High, low, and close bars	ThickLines may be used.
	1	No close bar	
	2	No high-low bars	
	3	No bars	

Data Type

Integer (Enumerated)

GraphTitle Property, Graph Control

Example

Places a text string above the graph.

Syntax

[*form*.]*Graph*.**GraphTitle**[:= *title*]

Remarks

A graph title cannot contain more than 80 characters.

A graph title may not be displayed if it is too long to fit on a graph. When this occurs, increase the width of the graph to display the graph title.

Data Type

String

GraphTitle Example, Graph Control

The following code places the title, "Title," at the top of a graph (GraphicsServer1) when you click a command button and no title currently exists. If the GraphTitle property does have a value, when you click the command button, the title will become blank. To try this example, paste the code into the Declarations section of a form that contains a command button and a graph.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
   GraphicsServer1.RandomData := 1;
   if  GraphicsServer1.GraphTitle := '' then
      GraphicsServer1.GraphTitle := 'Title'
   else
      GraphicsServer1.GraphTitle := '';
   GraphicsServer1.DrawMode := 2;
end;
```

GraphType Property, Graph Control

Specifies the type of graph. For illustrations of the different types of graphs, see the *Custom Control Reference*.

Syntax

[*form*.]*Graph*.**GraphType**[:= *setting*]

Remarks

The following table lists the GraphType property settings for the graph control.

Setting	Description
0	None
1	2D pie
2	3D pie
3	(Default) 2D bar
4	3D bar
5	Gantt
6	Line
7	Log/Lin
8	Area
9	Scatter
10	Polar
11	HLC

For each graph type there are many style options. For more information, see the GraphStyle property.

Data Type

Integer (Enumerated)

GridStyle Property, Graph Control

Places reference grids on the graph axes. For illustrations showing each style of grid, see the *Custom Control Reference*.

Syntax

[*form*.]*Graph*.**GridStyle**[:= *setting*]

The following table lists the GridStyle property settings for the graph control.

Setting	Description
0	(Default) None
1	Horizontal
2	Vertical
3	Both

For polar graphs, the horizontal axes are concentric circles, and the vertical axes are radial lines (spokes).

Data Type

Integer (Enumerated)

ImageFile Property, Graph Control

Sets a file name to which the bitmap or metafile is written when DrawMode is set to 6. If a path is not specified, the current directory is used.

Syntax

[*form*.]*Graph*.**ImageFile**[:= *filename*]

Remarks

The appropriate extension (.BMP or .WMF) is appended automatically. If you set DrawMode to 3 (Blit), a bitmap is created; otherwise, a metafile is created.

Note You cannot use this property to create a 256-color bitmap.

Data Type

String

IndexStyle Property, Graph Control

Example1 Example2

Sets the data array index style.

Syntax

[*form*.]*Graph*.**IndexStyle**[:= *setting*]

Remarks

The following table lists the IndexStyle property settings for the graph control.

Setting	Description
0	(Default) Standard. One-dimensional arrays are accessed through the ThisPoint property.
1	Enhanced. One-dimensional arrays are accessed through the IndexStyle property.

When IndexStyle = 1, the graph control's arrays are accessed as described in the following table.

Array	Properties used
GraphData	ThisSet and ThisPoint (two-dimensional array).
ColorData	ThisSet or ThisPoint.
ExtraData	ThisSet or ThisPoint.
LabelText	ThisPoint.
LegendText	ThisSet or ThisPoint.
PatternData	ThisSet or ThisPoint.
SymbolData	ThisSet.
XPosData	ThisSet and ThisPoint (two-dimensional array).

If the current graph type is a pie chart or a single-data-set bar graph, ThisPoint is used. For any other graph types, ThisSet is used. Pie charts and single-data-set bar graphs use ThisPoint because they display legends per point rather than per data set.

Note If the AutoInc property is on, the IndexStyle setting does not matter because AutoInc increments ThisSet and ThisPoint correctly irrespective of the IndexStyle setting. Also, once data arrays have been created, graphs are drawn in the normal way, regardless of the IndexStyle property.

Data Type

Integer (Enumerated)

IndexStyle Example 1, Graph Control

```
var
   I, J: integer;
begin
   GraphicsServer1.GraphType := 6    { Line graph };
   GraphicsServer1.IndexStyle := 1   { Enhanced index style };

   for I := 1 to GraphicsServer1.NumSets do
   begin
      GraphicsServer1.ThisSet := I;
      for J := 1 to GraphicsServer1.NumPoints do
      begin
         GraphicsServer1.ThisPoint := J;
         GraphicsServer1.GraphData := your data value;
         GraphicsServer1.XPosData := your data value;
      end;
   end;

   for I := 1 to GraphicsServer1.NumSets do
   begin
      GraphicsServer1.ThisSet := I   { Use ThisSet as index };
      GraphicsServer1.Leg endText := 'Data set' + IntToStr(I);
      GraphicsServer1.ExtraData := your data value;
      GraphicsServer1.ColorData := your data value;
      GraphicsServer1.PatternData := your data value;
      GraphicsServer1.SymbolData := your data value;
   end;

   for I := 1 to GraphicsServer1.NumPoints do
   begin
      GraphicsServer1.ThisPoint := I;
      GraphicsServer1.LabelText := 'Data point' := IntToStr(I);
   end;

   GraphicsServer1.DrawMode := 2;
end;
```

IndexStyle Example 2, Graph Control

```
GraphicsServer1.GraphType := 6    { Line graph };
GraphicsServer1.IndexStyle := 0 { Standard index style };

for I := 1 to GraphicsServer1.NumSets do
begin
   GraphicsServer1.ThisSet := I ;
   for J := 1 to GraphicsServer1.NumPoints do
   begin
      GraphicsServer1.ThisPoint := J;
      GraphicsServer1.GraphData := your data value;
      GraphicsServer1.XPosData := your data value;
   end;
end;

for I := 1 to GraphicsServer1.NumSets do
begin
   GraphicsServer1.ThisPoint := I { Use ThisPoint as index };
   GraphicsServer1.Leg end Text := 'Leg end' + IntToStr(I);
   GraphicsServer1.ExtraData := your data value;
   GraphicsServer1.ColorData := your data value;
   GraphicsServer1.PatternData := your data value;
   GraphicsServer1.SymbolData := your data value;
end;

for I := 1 to GraphicsServer1.NumPoints do
begin
   GraphicsServer1.ThisPoint := I;
   GraphicsServer1.LabelText := 'Label' := IntToStr(I);
end;

GraphicsServer1.DrawMode := 2;
```

LabelEvery Property, Graph Control

Determines the frequency of labels displayed on the X axis.

Syntax

[*form*.]*Graph*.**LabelEvery**[:= *frequency*]

Remarks

Enter a value between 1 (the default) and 1000, inclusive.

For example, suppose you have a graph with five points and the LabelText property is set to "Jan," "Feb," "Mar," "Apr," and "May." If the LabelEvery property is set to 1, all five labels are displayed. If it is set to 2, "Jan," "Mar," and "May" (the first, third, and fifth labels) are displayed. Finally, if LabelEvery is set to 3, only "Jan" and "Apr" (the first and fourth labels) are displayed.

Note The LabelEvery property only affects the graph control when the XPosData property is not set. Therefore, LabelEvery never affects scatter diagrams, which always use XPosData.

Data Type

Integer

Labels Property, Graph Control

Determines if labels are displayed along the graph's X and Y axes. For pie charts, this property determines if labels are displayed.

Syntax

[*form*.]*Graph*.**Labels**[:= *setting*]

Remarks

The following table lists the Labels property settings for the graph control.

Setting	Description
0	(Default) Off
1	On
2	X labels displayed
3	Y labels displayed

You can display the labels for the X and Y axes separately. This property operates independently of the Ticks property.

Data Type

Integer (Enumerated)

LabelText Property, Graph Control

Allows label text to be entered. For illustrations of this property, see the *Custom Control Reference*.

Syntax

[*form*.]*Graph*.**LabelText**[:= *label*]

Remarks

If no text has been entered, the labels show the value of the ThisPoint property for all graphs except pie charts, which show the magnitude of the slices.

Since this is an array property, the array element you set is determined by the current value of the ThisPoint property.

When entering text, you may use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new string, the ThisPoint counter is automatically incremented.

The LabelText property cannot contain more than 80 characters.

Label text may not be displayed if it is too long to fit on a graph.

Data Type

String

LeftTitle Property, Graph Control

Example

Places the text string that you provide to the left of the vertical axis.

Syntax

[*form*.]*Graph*.**LeftTitle**[:= *title*]

Remarks

This property is ignored for pie charts.

A left title cannot contain more than 80 characters.

A left title may not be displayed if it is too long to fit on a graph. When this occurs, increase the width of the graph to display the left title.

Data Type

String

LeftTitle Example, Graph Control

;

The following code places the title, "Title," to the left of the vertical axis of a graph (GraphicsServer1) when you click a command button and LeftTitle currently has no value. If the LeftTitle property does contain a text string, when you click the command button, the title will become blank. To try this example, paste the code into the Declarations section of a form that contains a command button and a graph.

```
procedure TForm1.Button1Click(Sender: TObject);
begin
   if  GraphicsServer1.LeftTitle := '' then
      GraphicsServer1.LeftTitle := 'Title';
   else
      GraphicsServer1.LeftTitle := '';
   GraphicsServer1.DrawMode := 2;
end;
```

LegendStyle Property, Graph Control

Gives the option of coloring the text you enter as legends (LegendText property). This color is in addition to the colored symbols or patterns.

Syntax

[*form*.]*Graph*.**LegendStyle**[:= *setting*]

Remarks

The following table lists the LegendStyle property settings for the graph control.

Setting	Description
0	Monochrome
1	Color

Data Type

Integer (Enumerated)

LegendText Property, Graph Control

Allows you to enter text for legends.

Syntax

[*form*.]*Graph.***LegendText**[:= *text*]

Remarks

There should be one text string for each data set. Pie charts and bar graphs with only one data set should have a string for each data point.

Since this is an array property, the array element is determined by the current value of the ThisPoint property.

When entering text, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new string, the ThisPoint counter is automatically incremented.

The LegendText property cannot contain more than 80 characters.

Legend text may not be displayed if it is too long to fit on a graph. When this occurs, increase the width of the graph to display the legend text.

Data Type

String

LineStats Property, Graph Control

Allows statistics lines to be superimposed on the graph. This property is valid for line or log/lin graphs only.

Syntax

[*form*.]*Graph*.**LineStats**[:= *setting*]

Remarks

The following table lists the LineStats property settings for the graph control.

Setting	Description
0	None.
1	Mean.
2	MinMax.
3	Mean and MinMax.
4	StdDev.
5	StdDev and Mean.
6	StdDev and MinMax.
7	StdDev and MinMax and Mean.
8	BestFit.
9	BestFit and Mean.
10	BestFit and MinMax.
11	BestFit and MinMax and Mean.
12	BestFit and StdDev.
13	BestFit and StdDev and Mean.
14	BestFit and StdDev and MinMax.
15	All.

Data Type

Integer (Enumerated)

NumPoints Property, Graph Control

Specifies the number of data points in each data set.

Syntax

[*form*.]*Graph*.**NumPoints**[:= *points*]

Remarks

The minimum value of NumPoints is 2. The default value for this property is 5.

The product of (NumPoints x NumSets) cannot be greater than 3800.

NumPoints can be changed at any time.

If NumPoints is less than the number of data items you have, excess array data is discarded. If NumPoints is greater than the number of data items you have, additional null-value data is created.

Data Type

Integer

NumSets Property, Graph Control

Specifies the number of data sets to be graphed.

Syntax

[*form*.]*Graph*.**NumSets**[:= *sets*]

Remarks

The minimum value for NumSets is 1. The default value for this property is 1.

The product of (NumPoints x NumSets) cannot be greater than 3800.

NumSets can be changed at any time.

If NumSets is less than the number of sets of data you have, any excess array data is discarded. If NumSets is greater than the number of data sets, additional null-value data is created.

Note Pie charts only use the first data set, even if NumSets > 1.

Data Type

Integer

Palette Property, Graph Control

Allows you to select a specific set of palette colors.

Syntax

[*form*.]*Graph*.**Palette** [:= *setting*]

Remarks

The following table lists the Palette property settings for the graph control.

Setting	Description
0	(Default) Solid
1	Pastel (dithered)
2	Grayscale (dithered)

If the Palette property is set to 1, the color values for the graph change from solid colors to dithered pastel colors. If the Palette property is set to 2, the color values for the graph are changed to the nearest dithered shade of gray equivalent.

Data Type

Integer (Enumerated)

PatternData Property, Graph Control

Selects a pattern for solid fills, a line pattern for patterned lines, or a line width (in pixels) for thick lines.

Syntax

[*form*.]*Graph*.**PatternData**[:= *pattern*]

Remarks

The PatternData property settings are illustrated in the following figure.

Pattern data values range from 0 to 31. Select one pattern per data set or one pattern per point for pie or bar charts with NumSets = 1.

For illustrations of the PatternData property settings, see the *Custom Control Reference*.

Since this is an array property, the array element you set is determined by the current value of the ThisPoint property.

When you enter data, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new value, the ThisPoint counter is automatically incremented.

Note Fill patterns 8 through 15 do not exist.

Data Type

Integer (Enumerated)

PatternedLines Property, Graph Control

Sets the style of the lines connecting the data points.

Syntax

[*form*.]*Graph*.**PatternedLines**[:= *setting*]

Remarks

The following table lists the PatternedLines property settings for the graph control.

Setting	Description
0	(Default) Off
1	On

When you set the PatternedLines property to 1 (on), the graph is plotted with dotted lines of pattern 1, unless a different PatternData has been set. For information on different pattern styles, see the PatternData property.

Data Type

Integer

Picture Property, Graph Control

Example

Passes a graph image directly to a picture control. This property is not available at design time and is read-only at run time.

Syntax

[*form*.]*Graph*.**Picture**

Data Type

Integer

Picture Example, Graph Control

The following code puts a copy of the graph currently displayed in GraphicsServer1 into Picture1.

```
;
Picture1.Picture := GraphicsServer1.Picture;
;
```

If Picture1 has a different aspect ratio from GraphicsServer1, the graph image is stretched or compressed accordingly.

PrintStyle Property, Graph Control

Selects the print style options when printing the control (DrawMode = 5).

Syntax

[*form*.]*Graph*.**PrintStyle**[:= *style*]

Remarks

The following table lists the PrintStyle property settings for the Graph control.

Setting	Description
0	(Default) Monochrome
1	Color
2	Monochrome with border
3	Color with border

The default option temporarily converts the DrawStyle to Monochrome (0) before printing. If you are using a color printer, or have a printer capable of printing gray scales, set PrintStyle = 1.

If you use these options with DrawMode = 5, the graph is printed with the best resolution of your printer. No bitmap is generated.

Data Type

Integer (Enumerated)

RandomData Property, Graph Control

If you set the RandomData property to 1 (on), it generates random data to be graphed. This is mainly useful at design time, when you want to see how the graph will appear at run time.

Syntax

[*form*.]*Graph*.**RandomData**[:= *setting*]

Remarks

The following table lists the RandomData property settings for the graph control.

Setting	Description
0	Off
1	(Default) On

Random numbers that are generated are never negative. To see the effect of negative values, enter your own data.

Note The RandomData property is automatically set to 0 (off) if GraphData values are present. You can override the GraphData values by setting the RandomData property to 1 (on). Setting it to 0 (off) again reinstates the GraphData values. Using DataReset with GraphData (or all data) sets the RandomData property back to 1 (on).

Data Type

Integer

SeeThru Property, Graph Control

Example

If you set the SeeThru property to 1 (on), the graph background is not cleared. Instead, whatever was there before you inserted the graph will show through. You can create special effects by drawing a graph over a picture control containing a bitmap. This property is available at run time only.

Syntax

[*form*.]*Graph*.**SeeThru**[:= *setting*]

Remarks

The following table lists the SeeThru property settings for the graph control.

Setting	Description
0	(Default) Off
1	On

To function correctly, some programming is necessary. Otherwise, the graph cannot be redrawn if it is covered and then uncovered by another window.

Note See-through graphs do not work when DrawMode = 3 (Blit).

Data Type

Integer

SeeThru Example, Graph Control

Create a paintbox (PaintBox1), and then create a graph (GraphicsServer1), not as a child of the picture, but directly on your form. Move the graph over the top of the picture, making sure the graph does not entirely cover the picture. Leave a narrow border all the way around to ensure the picture receives paint messages. The BorderStyle should be set to None, or a black line will appear around the area of the graph.

When the paintbox (PaintBox1) receives a paint message, it refreshes both itself and the graph (GraphicsServer1), ensuring that the graph is still on top of the picture with the picture showing through. The flag is necessary to prevent entering the loop again. The Paint event is triggered by the OnPaint event.

```
var
   Flag: integer;

procedure TForm1.FormCreate(Sender: TObject);
begin
   Flag := 0;
   GraphicsServer1.SeeThru := 1;
end;

procedure TForm1.PaintBox1Paint(Sender: TObject);
begin
   if  Flag := 1 then
      begin
      Flag := 0;
      Picture1.Refresh;
      GraphicsServer1.Refresh;
      end;
   else
      Flag := 1;
end;
```

SymbolData Property, Graph Control

Selects symbols to be used for line, log/lin, scatter, and polar graphs.

Syntax

[*form*.]*Graph*.**SymbolData**[:= *symbol*]

Remarks

The following table describes the settings for the SymbolData property

Setting	Description
0	Cross (+)
1	Cross (X)
2	Triangle (up)
3	Solid Triangle (up)
4	Triangle (down)
5	Solid Triangle (down)
6	Square
7	Solid Square
8	Diamond
9	Solid Diamond

You should select one symbol per data set. The default setting is 0.

Since this is an array property, the array element you set is determined by the current value of the ThisPoint property.

When you enter data, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new value, the ThisPoint counter is automatically incremented.

Data Type

Integer (Enumerated)

ThickLines Property, Graph Control

Sets the width of the lines. For illustrations, see the *Custom Control Reference*.

Syntax

[*form*.]*Graph*.**ThickLines**[:= *setting*]

Remarks

The following table lists the ThickLines property settings for the graph control.

Setting	Description
0	(Default) Off
1	On

When the ThickLines property is set to 1 (on), 3-pixel-thick lines are drawn, unless a PatternData property is set. If DrawStyle = 0 (Monochrome), line widths between 2 and 7 pixels (depending on the PatternData property setting) are selected.

Data Type

Integer

ThisPoint Property, Graph Control

Example

Sets the current point number manually so that a particular data point can be changed.

Syntax

[*form*.]*Graph*.**ThisPoint**[:= *point*]

Remarks

The property settings for ThisPoint are from 1 to NumPoints. Setting ThisPoint overrides the AutoInc setting.

Data Type

Integer

ThisPoint Example, Graph Control

The following code draws a 3D bar graph with 1 data set and 5 points. To try this example, paste this code into the Form_Load event procedure of a form that contains a graph (GraphicsServer1).

```
procedure TForm1.FormCreate(Sender: TObject);
var
   I: integer;
begin
   GraphicsServer1.NumPoints := 5;
   GraphicsServer1.NumSets := 1;
   GraphicsServer1.AutoInc := 1;
   for I := 1 to 5 do
      GraphicsServer1.GraphData := I;
   GraphicsServer1.ThisPoint := 3;
   GraphicsServer1.GraphData := 10;
   GraphicsServer1.GraphType := 4;
   GraphicsServer1.DrawMode := 2;
end;
```

ThisSet Property, Graph Control

Example

Allows you to manually control the current set number so that a particular data set can be changed.

Syntax

[*form*.]*Graph*.**ThisSet**[:= *set*]

Remarks

The property settings for ThisSet are from 1 to NumSets. Setting ThisSet overrides the AutoInc setting. This allows you to address any individual data point when you have multiple data sets.

Data Type

Integer

ThisSet Example, Graph Control

The following code draws a 3D bar graph with 3 data sets with 5 points in each set. To try this example, paste this code into the Form_Load event procedure of a form that contains a graph (GraphicsServer1).

```
procedure TForm1.FormCreate(Sender: TObject);
var
   I: integer;
begin
   GraphicsServer1.NumPoints := 5;
   GraphicsServer1.NumSets := 3;
   GraphicsServer1.AutoInc := 1;
   for I := 1 to GraphicsServer1.NumPoints * GraphicsServer1.NumSets do
      GraphicsServer1.GraphData := 5;
   GraphicsServer1.ThisSet := 2;
   GraphicsServer1.ThisPoint := 3;
   GraphicsServer1.GraphData := 10;
   GraphicsServer1.GraphType := 4;
   GraphicsServer1.DrawMode := 2;
end;
```

TickEvery Property, Graph Control

Determines the interval between tick marks on the X axis. The TickEvery value specifies that the tick mark represents *n* data points, where *n* is a value in the range 1 to 1000. The default value for this property is 1.

Syntax

[*form*.]*Graph*.**TickEvery**[:= *interval*]

Remarks

This property is ignored when the XPosData property is set. This means that the TickEvery property never has any effect on scatter graphs, which always have XPosData property values.

If the NumPoints property is less than TickEvery, the X axis of your graph is extended to the value of TickEvery. Also, since there must always be an integral number of ticks, the X axis will be extended to a multiple of TickEvery, if necessary. For example, if NumPoints = 127 and TickEvery = 50, then the X axis is extended to 150.

Data Type

Integer

Ticks Property, Graph Control

Determines whether axis ticks are displayed.

Syntax

[*form*.]*Graph*.**Ticks**[:= *setting*]

Remarks

You can turn ticks on and off separately for the X and Y axes.

This property operates independently of the Labels property. Ticks has no affect on a three-dimensional graph drawn with a cage affect.

The following table lists the Ticks property settings for the graph control.

Setting	Description
0	(Default) Off
1	On
2	X ticks
3	Y ticks

Data Type

Integer (Enumerated)

XPosData Property, Graph Control

Example

Gives an independent X value for a graph.

Syntax

[*form*.]*Graph*.**XPosData**[:= *xvalue*]

Remarks

The property setting for XPosData is any real number.

This property can be set for all graph types except pie and Gantt charts.

Since this is a two-dimensional array property, the array element you set is determined by the current value of the ThisSet and ThisPoint properties.

When you enter data, you can use the AutoInc property. If you set the AutoInc property to 1 (on), every time you set a new value, the ThisSet and ThisPoint counters are automatically incremented.

If you have multiple sets of GraphData, but only one set of XPosData, the graph control automatically applies the single set of XPosData to each set of GraphData.

Data Type

Single

XPosData Example, Graph Control

```
procedure TForm1.FormCreate(Sender: TObject);
var
   I, J: integer;
begin
   GraphicsServer1.AutoInc := 0;
   GraphicsServer1.NumPoints := 10;
   GraphicsServer1.NumSets := 2;
   for I := 1 to 2 do
   begin
      GraphicsServer1.ThisSet := I;
      for J := 1 to 10 do
      begin
         GraphicsServer1.ThisPoint := J;
         if  I := 1 then GraphicsServer1.GraphData := 5 - J;
         if  I := 2 then GraphicsServer1.GraphData := J - 5;
         GraphicsServer1.XPosData := J;
      end;
   end;
   GraphicsServer1.DrawMode := 2;
end;
```

YAxisMax, YAxisMin Properties, Graph Control

Specifies the maximum Y-axis value (YAxisMax) and minimum Y-axis value (YAxisMin) on your graph.

Syntax

[*form*.]*Graph*.**YAxisMax**[:= *max*]

[*form*.]*Graph*.**YAxisMin**[:= *min*]

Remarks

The property settings for YAxisMax and YAxisMin are any real numbers.

These properties are used in combination with YAxisTicks and only take affect when YAxisStyle = 2 (user-defined). For more information, see the YAxisStyle property.

Data Type

Single

YAxisPos Property, Graph Control

Specifies the position of the Y axis on your graph.

Syntax

[*form*.]*Graph*.**YAxisPos**[:= *position*]

Remarks

The following table lists the YAxisPos property settings for the graph control.

Setting	Description
0	(Default) Y axis is positioned automatically according to your XPosData values. When the values are all positive, the Y axis appears at the leftmost edge of the graph. If the values are all negative, the Y axis appears on the rightmost edge of the graph.
1	Left.
2	Right.

Data Type

Integer (Enumerated)

YAxisStyle Property, Graph Control

Specifies the method used to scale and range the Y axis on your graph.

Syntax

[*form*.]*Graph*.**YAxisStyle**[:= *style*]

Remarks

The following table lists the YAxisStyle property settings for the graph control.

Setting	Description
0	(Default) Y-axis range is calculated automatically based on the data to be graphed. The maximum Y-axis value is greater than or equal to the maximum data value. The minimum axis value is 0, or, if the data includes negative values, it is less than or equal to the minimum data value. The Y axis, therefore, always includes the 0 origin.
1	Variable origin. The maximum Y-axis value is equal to or greater than the maximum data value. The minimum Y-axis value is less than or equal to the minimum data value, whether the data includes negative values or not. The Y axis, therefore, may not include the 0 origin.
2	User-defined origin. The YAxisMax, YAxisMin, and YAxisTicks properties work together to control the range.

The variable origin style is useful when you are graphing data with a small variation around a nonzero value. If you use the default style, the variation may not be visible.

Use the user-defined style when you want to present the data in a certain way. For example, to create a series of comparable graphs, you might set the Y-axis range from 1000 to +1000, even though the data values for some graphs are all positive.

Caution If your data exceeds the limits of the Y-axis range, the graph is drawn outside of the axes bounds and can result in strange effects.

YAxisTicks specifies the number of ticks from the origin to the greater of the YAxisMax and YAxisMin values, regardless of sign. Because there must always be an integral number of ticks on an axis, the graph will sometimes override the YAxisMin value or YAxisMax value.

In this example, YAxisMax has the greater value: YAxisMax = 300, YAxisMin = 10, and YAxisTicks = 3. The graph places ticks 100 units apart, and the YAxisMin value displayed is 100.

In this example, YAxisMin has the greater value (even though it is negative): YAxisMax = 10, YAxisMin = 300, and YAxisTicks = 3. The YAxisMax value displayed is 100.

Data Type

Integer (Enumerated)

YAxisTicks Property, Graph Control

Specifies the number of ticks on the Y axis of your graph.

Syntax

[*form*.]*Graph*.**YAxisTicks**[:= *ticks*]

Remarks

Enter a value between 1 (default) and 100, inclusive.

YAxisTicks works in combination with YAxisMax and YAxisMin and is only used when YAxisStyle = 2 (user-defined). For more information, see the YAxisStyle property.

Data Type

Integer

Graph Control Extended Version

The extended version of the Graph control includes the following addition features:

- Rotating graphs
- Hot events for drill-down
- Combo graphs
- Curve fitting
- More graph types
- Extended customization

For more information on the extended version of the Graph control, please complete this form and mail or fax it to one of the publishers below.

USA & International
Pinnacle Publishing Inc
PO Box 888, Kent,
WA 98035-0888, USA
Tel: 206/251-1900
Fax: 206/251-5057

Germany & Austria
heilerSoftware
Mittlerer Pfad 5
70499 Stuttgart
Germany
Tel: 0711 139840
Fax: 0711 8666301

UK & rest of Europe
Bits Per Second Ltd
14 Regent Hill, Brighton,
Sussex BN1 3ED, UK
Tel: 01273 727119
Fax: 01273 731925

NAME _____

COMPANY _____

ADDRESS _____

_____

_____

FAX _____

PHONE _____

Graph Control Constants

AutoInc Constants

Constant	Value	Description
gphOff	0	Automatic incrementing off.
gphOn	1	Automatic incrementing on.

DrawLineStyle and LegendStyle Constants

Constant	Value	Description
gphMonochrome	0	Sets background white and all colors black.
gphColor	1	Uses specified colors.

GraphType Constants

Constant	Value	Description
gphNone	0	No graph.
gphPie2d	1	Two-dimensional pie chart.
gphPie3d	2	Three-dimensional pie chart.
gphBar2d	3	Two-dimensional bar chart.
gphBar3d	4	Three-dimensional bar chart.
gphGantt	5	Gantt chart.
gphLine	6	Line graph.
gphLogLin	7	Log/Lin graph.
gphArea	8	Area graph.
gphScatter	9	Scatter graph.
gphPolar	10	Polar graph.
gphHLC	11	High-low-close graph.

BackgroundColor, ForegroundColor, and ColorData Constants

Constant	Value	Description
gphBlack	0	Black.
gphBlue	1	Blue.
gphGreen	2	Green.
gphCyan	3	Cyan.
gphRed	4	Red.
gphMagenta	5	Magenta.
gphBrown	6	Brown.
gphLightGray	7	Light gray.
gphDarkGray	8	Dark gray.
gphLightBlue	9	Light blue.
gphLightGreen	10	Light green.
gphLightCyan	11	Light cyan.
gphLightRed	12	Light red.
gphLightMagenta	13	Light magenta.
gphYellow	14	Yellow.
gphWhite	15	White.
gphAutoBW	16	(Default) Automatic black and white. Only available in Foreground constants.

SymbolData Constants

Constant	Value	Description
gphCrossPlus	0	Plus sign (+) symbol.
gphCrossTimes	1	Multiplication sign (x) symbol.
gphTriangleUp	2	Upright triangle symbol.
gphSolidTriangle	3	Solid triangle symbol.
gphTriangleDown	4	Upside-down triangle symbol.
gphSolidTriangle	5	Solid triangle symbol.
gphSquare	6	Square symbol.
gphSolidSquare	7	Solid square symbol.
gphDiamond	8	Diamond symbol.
gphSolidDiamond	9	Solid diamond symbol.

GridStyle Constants

Constant	Value	Description
gphNone	0	No grid.
gphHorizontal	1	Horizontal grid.
gphVertical	2	Vertical grid.
gphBoth	3	Both grids.

DataReset Constants

Constant	Value	Description
gphNone	0	No reset.
gphGraphData	1	Resets graph data.
gphColorColorData	2	Resets color data.
gphExtraData	3	Resets extra data.
gphLabelText	4	Resets label text.
gphLegendText	5	Resets legend text.
gphPatternData	6	Resets pattern data.
gphSymbolData	7	Resets symbol data.
gphXPosData	8	Resets x-position data.
gphAllData	9	Resets all data.
gphFontInfo	10	Resets font information.

DrawMode Constants

Constant	Value	Description
gphNoAction	0	No graph drawn at design time.
gphClear	1	No graph drawn at design time, but properties displayed.
gphDraw	2	Displays graph at design time and run time.
gphBlit	3	Displays graph using blitting technique.
gphCopy	4	Copies graph to Clipboard.
gphPrint	5	Sends graph to printer.
gphWrite	6	Writes graph to disk.

FontStyle Constants

Constant	Value	Description
gphDefault	0	Default.
gphItalic	1	Italic.
gphBold	2	Bold.
gphBoldItalic	3	BoldItalic.
gphUnderlined	4	Underlined.
gphUnderlinedItalic	5	UnderlinedItalic.
gphUnderlinedBold	6	UnderlinedBold.
gphUnderlinedBoldItalic	7	UnderlinedBoldItalic.

FontFamily Constants

Constant	Value	Description
gphRoman	0	Roman.
gphSwiss	1	Swiss.
gphModern	2	Modern.

FontUse Constants

Constant	Value	Description
gphGraphTitle	0	GraphTitle.
gphOtherTitles	1	OtherTitles.
gphLabels	2	Labels.
gphLegend	3	Legend.
gphAllText	4	AllText.

IndexStyle Constants

Constant	Value	Description
gphStandard	0	Standard.
gphEnhanced	1	Enhanced.

Labels Constants

Constant	Value	Description
gphOff	0	Off.
gphOn	1	On.
gphXAxisLabelsOnly	2	X-axis labels only.
gphYAxisLabelsOnly	3	Y-axis labels only.

LineStats Constants

Constant	Value	Description
gphNone	0	None.
gphMean	1	Mean.
gphMinmax	2	MinMax.
gphMeanMinmax	3	Mean and MinMax.
gphStddev	4	Stddev.
gphStddevMean	5	StdDev and Mean.
gphStddevMinmax	6	StdDev and MinMax.
gphStddevMinmaxMean	7	StdDev and MinMax and Mean.
gphBestfit	8	BestFit.
gphBestfitMean	9	BestFit and Mean.
gphBestfitMinmax	10	BestFit and MinMax.
gphBestfitMinmaxMean	11	BestFit and MinMax and Mean.
gphBestfitStddev	12	BestFit and StdDev.
gphBestfitStddevMean	13	BestFit and StdDev and Mean.
gphBestfitStddevMinmax	14	BestFit and StdDev and MinMax.
gphAll	15	All.

Palette Constants

Constant	Value	Description
gphDefault	0	Default.
gphPastel	1	Pastel.
gphGrayscale	2	Grayscale.

PatternedLines Constants

Constant	Value	Description

gphPatternOff	0	Pattern off.
gphPatternOn	1	Pattern on.

PrintStyle Constants

Constant	Value	Description
gphMonochrome	0	Color.
gphColor	1	Color with border.
gphMonochromeWithBorder	2	Monochrome.
gphColorWithBorder	3	Monochrome with border.

RandomData Constants

Constant	Value	Description
gphOff	0	Off.
gphOn	1	On.

ThickLines Constants

Constant	Value	Description
gphLinesOff	0	Lines off.
gphLinesOn	1	Lines on.

YAxisPos Constants

Constant	Value	Description
gphDefault	0	Default.
gphAlignLeft	1	Align left.
gphAlignRight	2	Align right.

YAxisStyle Constants

Constant	Value	Description
gphDefault	0	Default.
gphVariableOrigin	1	Variable origin.
gphUserDefined	2	User-defined.

Ticks Constants

Constant	Value	Description
gphTicksOff	0	Ticks off.
gphTicksOn	1	Ticks on.
gphXAxisTicksOnly	2	X-axis ticks only.
gphYAxisTicksOnly	3	Y-axis ticks only.