Embedded Controls for Microsoft Business Solutions

(Technical: ChartSpace methods and properties)

DISCLAIMER

COPYRIGHT NOTICE

TRADEMARKS

Published by Logic Vision B.V., Netherlands.

Published in Netherlands 2003.

DocID: Embedded Controls MBS (technical: chartspace methods and properties)

ChartSpace methods and properties glossary

The “Manual Embedded Controls MBS (technical).pfd” document explains you the basics of displaying an Embedded Control on a form and Report. This is enough to display simple charts but the “Embedded ChartSpace Control” has a lot more you might be interested to use. This document explains it’s entire interface.

Subroutine ShowGraph

<no parameters>

The ShowGraph function creates a new chart in the control. If a chart already exists then the this chart will be removed. Only 1 chart can be displayed in one control instance.

The ShowGraph Function can be called prior to setting matrix data and properties but also after setting data and properties. If ShowGraph is called afterwards setting properties and data then all preset data and properties are immediately applied.

Subroutine ClearGraph

<no parameters>

The ClearGraph function removes the chart from the control and also clears the matrix where the data is stored as well as legend and category data.

However all display properties will remain until the next time ShowGraph is called.

The “Title” property will be cleared.

Subroutine SetCell

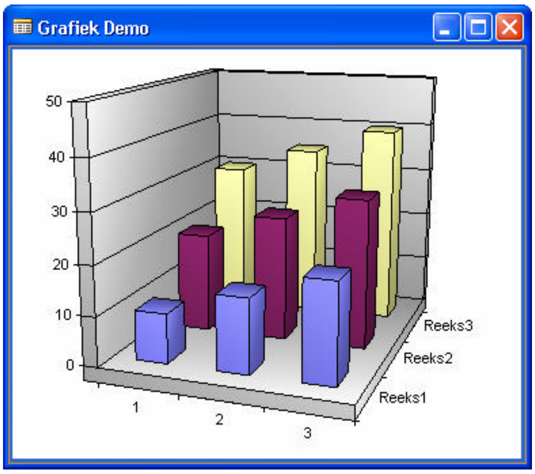
x <Integer>
y <Integer>
number <Decimal>

The subroutine SetCell will add a value to the matrix a graph consist of. By giving the *x* position (legend) and the *y* position (category), the value *number* will be entered in the matrix at the *x,y* position.

The *x* variable has a maximum of 100 elements (0-99). The y variable has a maximum of 10000 elements (0-9999).

The *number* variable has a range of -1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values.

The Matrix size will automatically grow as required. Therefore for performance reason, if the number of elements is known and quite large, the best thing to do is enter the highest *x,y* element first. If the value of the highest *x,y* element is not available at that time set it to 0 first and overwrite it at a later point in the routine. This allows the control to allocate memory once instead of every time the number of elements are extended.



Example

```
ChartSpaceControl.SetCell (2,2,0);  // Calibrate matrix to 3x3 elements
ChartSpaceControl.SetCell (0,0,10);
ChartSpaceControl.SetCell (0,1,15);
ChartSpaceControl.SetCell (0,2,20);
ChartSpaceControl.SetCell (1,0,20);
ChartSpaceControl.SetCell (1,1,25);
ChartSpaceControl.SetCell (1,2,30);
ChartSpaceControl.SetCell (2,0,30);
ChartSpaceControl.SetCell (2,1,35);
ChartSpaceControl.SetCell (2,2,40);
```

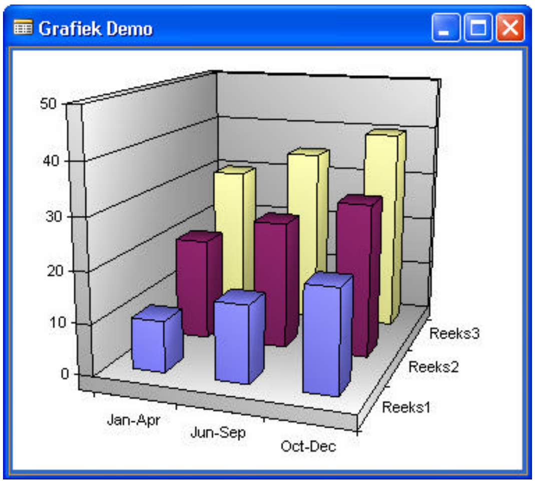
The matrix will look like this (never visible to programmer or user):

10	15	20
20	25	30
30	35	40

Subroutine SetCategory

x <Integer>
value <Text>

The SetCategory subroutine enables us to fill the category (x) axis. The number *x* represents the x category's in the matrix that is filled by SetCell. The text *value* represents the text the x range represents. This could for example be a date when displaying time charts.



When no category is set the chart will automatically number the categories. However there would be very little cases this would be acceptable.

If one category axis value is not set this category will not be automatically numbered.

If 2 category axis values are set exactly the same the last category axis will not be shown. So the categories axis values should always be differ from each other.

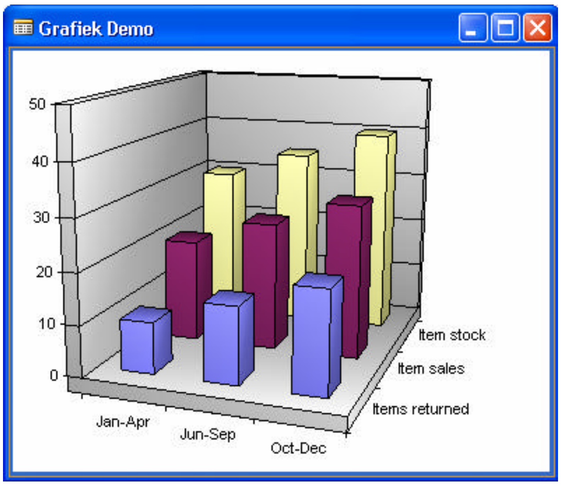
Example

```
ChartSpaceControl.SetCategory (0, 'Jan-Apr');
ChartSpaceControl.SetCategory (1, 'Jun-Sep');
ChartSpaceControl.SetCategory (2, 'Oct-Dec');
```

Subroutine SetLegend

y <Integer>
value <Text>

The subroutine SetLegend works exactly the same as the SetCategory subroutine except it is filling the filling the ´y´ category axis or legend.



In case of a 2D chart only the legend values can be set with this function. However the hasLegend property must be set to TRUE.

In case of a 3D chart both the legend and the y category axis (depth axis) values will be set with this function. A legend then is not necessary so the property hasLegend can be set to FALSE.

Example:

```
ChartSpaceControl.SetLegend (0, 'Items returned');
ChartSpaceControl.SetLegend (1, 'Item sales');
ChartSpaceControl.SetLegend (2, 'Item stock');
```

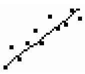
Subroutine SetTrendLine

y <Integer>
trendtype <Integer>

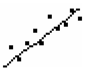
The SetTrendLine subroutine adds another range to the chart, but you won't have to enter data for this range, the chart calculates the values by a mathematical formula. A trendline can be attached to a range *y* and can have 5 different types.

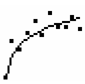
The SetTrendLine subroutine must be called after the ShowGraph subroutine is called and a Graph is visible. It is also important that the *l* range exists. If either of this is not the case the call to the SetTrendLine subroutine will be omitted.

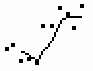
The 6 different types are:

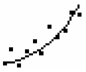

0 - Polynomial

1 - Exponential

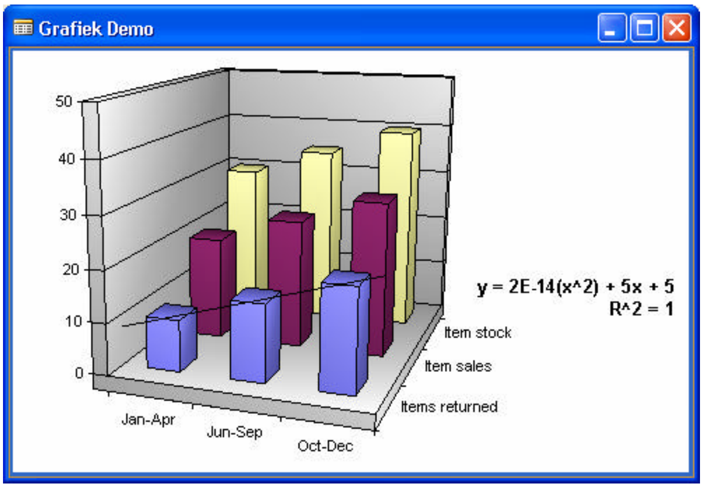

2 - Linear


3 - Logarithmic


4 - Moving Average


5 - Power

Example:
`ChartSpaceControl.SetTrendLine (0, 0);`



Subroutine SetSerieColor

y <Integer>
color <Text>

By default the ChartSpace automatically assigns a color to a serie. In some cases you may want to decide by yourself what color a serie should have. Use the subroutine SetSerieColor to assign a custom color to a serie. To assign a color you can use a word like 'Red' or use the color number like '012345'.

When using a number use for the parameter *color* the value '0' to '16777215' (note that this is not an integer variable but a text variable used as a variant).

Use a color number for the parameter *color* or one of the values in this table:

ALICEBLUE	ANTIQUEWHITE	AQUA	AQUAMARINE
AZURE	BEIGE	BISQUE	BLACK
BLANCHEDALMOND	BLUE	BLUEVIOLET	BROWN
BURLYWOOD	CADETBLUE	CHARTREUSE	CHOCOLATE
CORAL	CORNFLOWER	CORNSILK	CRIMSON
CYAN	DARKBLUE	DARKCYAN	DARKGOLDENROD
DARKGRAY	DARKGREEN	DARKKHAKI	DARKMAGENTA
DARKOLIVEGREEN	DARKORANGE	DARKORCHID	DARKRED
DARKSALMON	DARKSEAGREEN	DARKSLATEBLUE	DARKSLATEGRAY
DARKTURQUOISE	DARKVIOLET	DEEPPINK	DEEPSKYBLUE
DIMGRAY	DODGERBLUE	FIREBRICK	FLORALWHITE
FORESTGREEN	FUCHIA	GAINSBORO	GHOSTWHITE
GOLD	GOLDENROD	GRAY	GREEN
GREENYELLOW	HONEYDEW	HOTPINK	INDIANRED
INDIGO	IVORY	KHAKI	LAVENDER
LAVENDERBLUSH	LAWNGREEN	LEMONCHIFFON	LIGHTBLUE
LIGHTCORAL	LIGHTCYAN	LIGHTGOLDENRODYELLOW	LIGHTGREEN
LIGHTGREY	LIGHTPINK	LIGHTSALMON	LIGHTSEAGREEN
LIGHTSKYBLUE	LIGHTSLATEGRAY	LIGHTSTEELBLUE	LIGHTYELLOW
LIME	LIMEGREEN	LINEN	MAGENTA
MAROON	MEDIUMAQUAMARINE	MEDIUMBLUE	MEDIUMORCHID
MEDIUMPURPLE	MEDIUMSEAGREEN	MEDIUMSLATEBLUE	MEDIUMSPRINGGREEN
MEDIUMTURQUOISE	MEDIUMVIOLETRED	MIDNIGHTBLUE	MINTCREAM
MISTYROSE	MOCCASIN	NAVAJOWHITE	NAVY
OLDLACE	OLIVE	OLIVEDRAB	ORANGE
ORANGERED	ORCHID	PALEGOLDENROD	PALEGREEN
PALETURQUOISE	PALEVIOLETRED	PAPAYAWHIP	PEACHPUFF
PERU	PINK	PLUM	POWDERBLUE
PURPLE	RED	ROSYBROWN	ROYALBLUE
SADDLEBROWN	SALMON	SANDYBROWN	SEAGREEN
SEASHELL	SIENNA	SILVER	SKYBLUE
SLATEBLUE	SLATEGRAY	SNOW	SPRINGGREEN
STEELBLUE	TAN	TEAL	THISTLE
TOMATO	TURQUOISE	VIOLET	WHEAT

WHITE	WHITESMOKE	YELLOW	YELLOWGREEN

Use the *y* parameter to specify the serie.

Subroutine SetSerieWeight

y <Integer>
Weight <Text>

With the subroutine SetSerieWeight you can specify how ‘thick’ the serie line is displayed.

Use the *Weight* parameter to specify the thickness of the serie line.
Use the *y* parameter to specify the serie.

If you don’t specify a weight for a serie then the standard value 2 is used.

Propety AxisMinimum <Decimal>

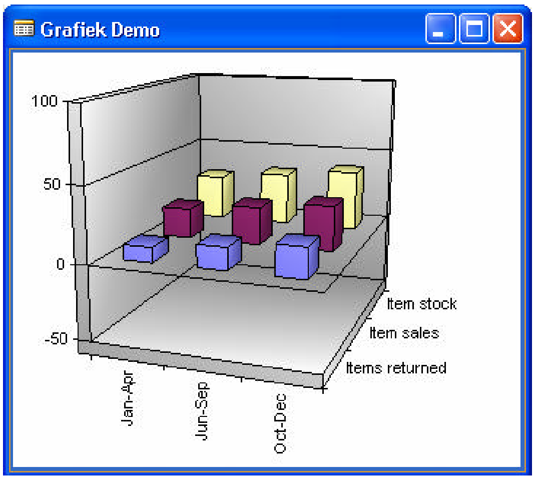
See AxisMaximum.

Property AxisMaximum <Decimal>

By default the minimum and maximum value of the axis is automatically generated by a rounding procedure done with the minimum and maximum value in the chart. In some cases you might want to overrule this automatically generated values. This overruling can be done with the AxisMinimum and AxisMaximum property.

Example:

```
ChartSpaceControl.AxisMinimum := -50;
ChartSpaceControl.AxisMaximum := 100;
```

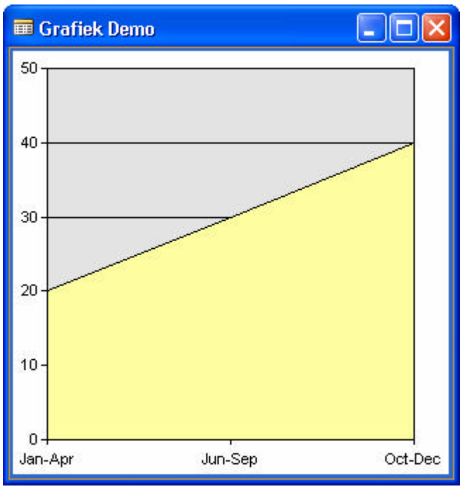


Property GraphType <Integer>

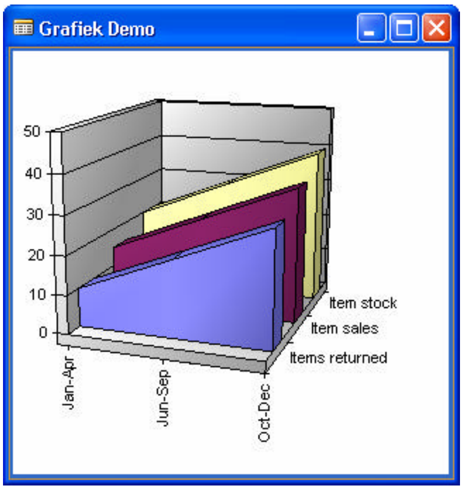
The property GraphType let’s you select the type of chart that is best suitable for your purpose. Most of the chart type’s in the “Microsoft Office Web Charts” are supported.

Here is a list with examples of different Chart types.

1 Area

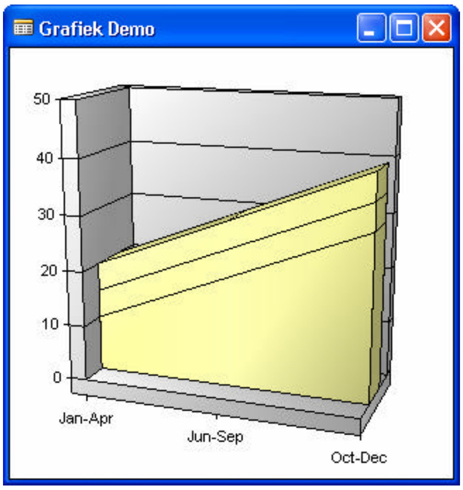


2 Area 3D

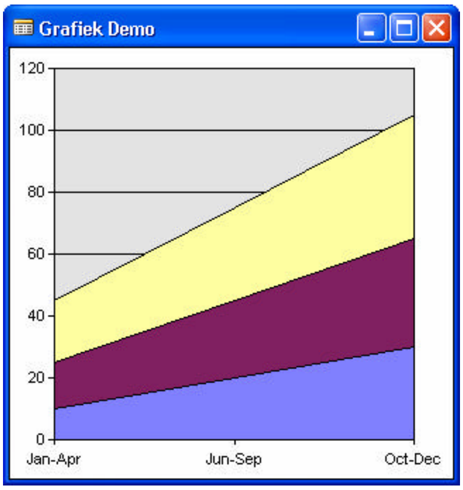


3 Area Overlapped 3D

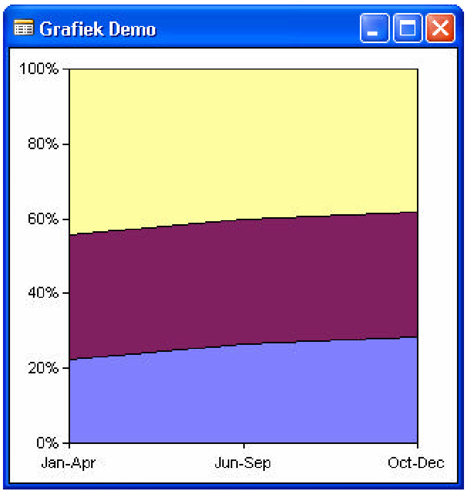
4 Area Stacked



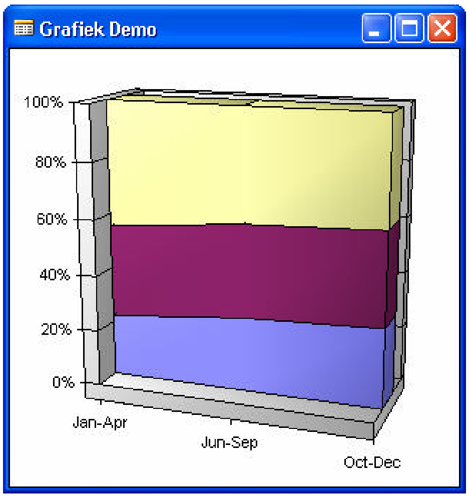
5 Area Stacked 100



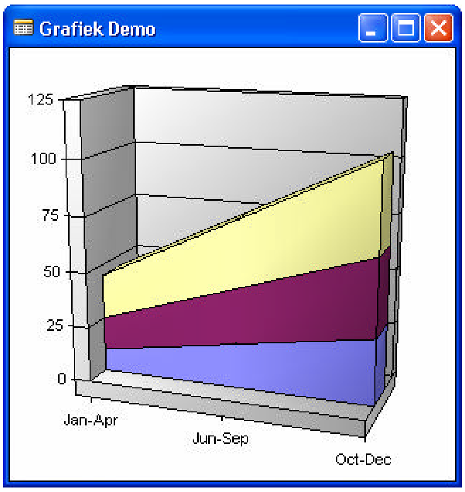
6 Area Stacked 100 3D



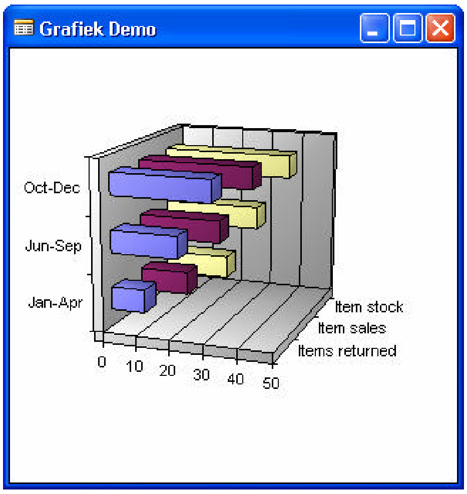
7 Area Stacked 3D



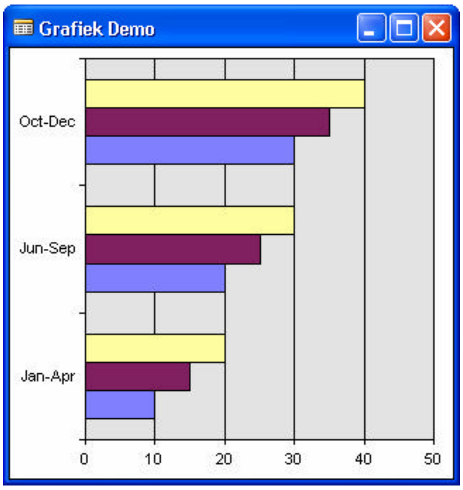
8 Bar 3D



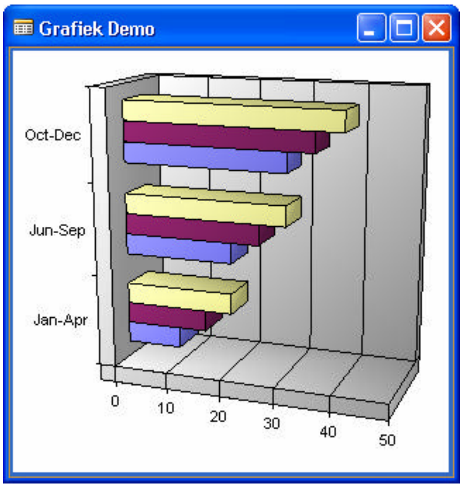
9 Bar Clustered



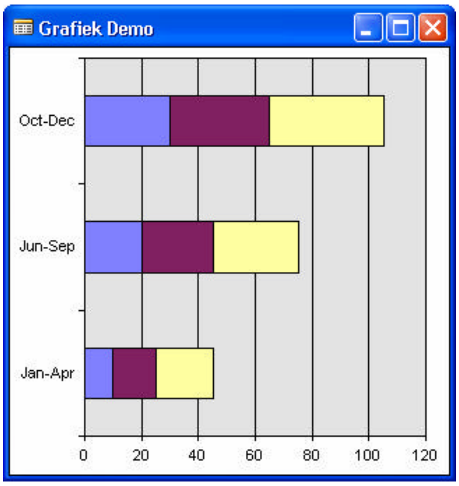
10 Bar Clustered 3D

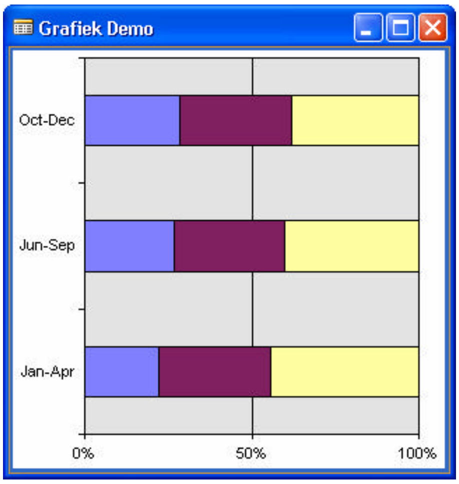


11 Bar Stacked

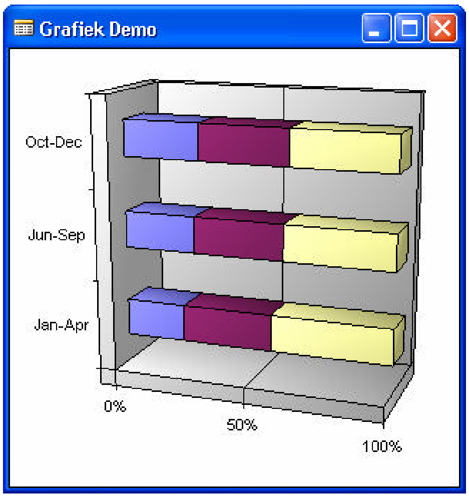


12 Bar Stacked 100

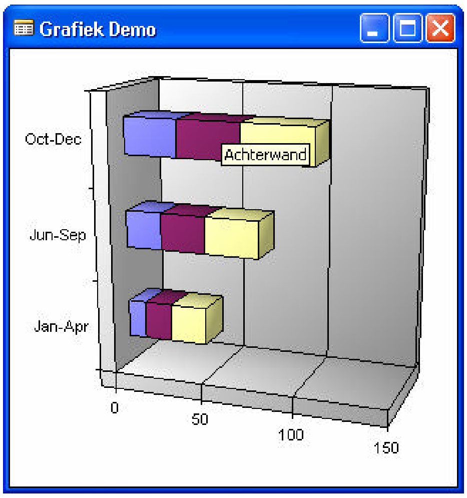




13 Bar Stacked 100 3D



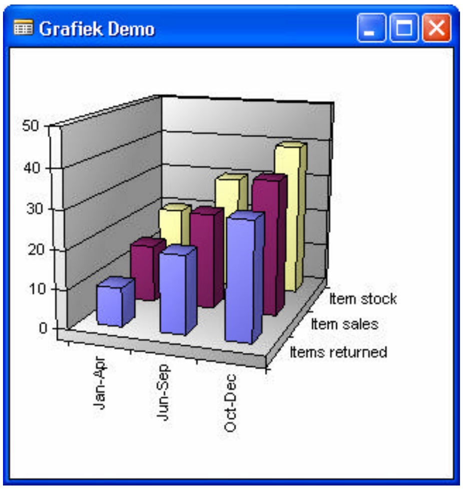
14 Bar Stacked 3D



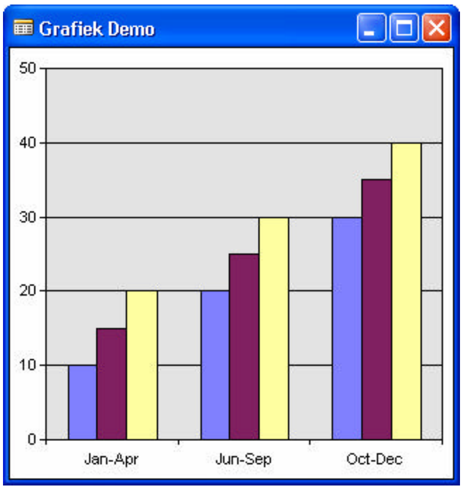
15 Bubble
Sorry, this chart type is not implemented.

16 Bubble 3D
Sorry, this chart type is not implemented.

17 Column 3D

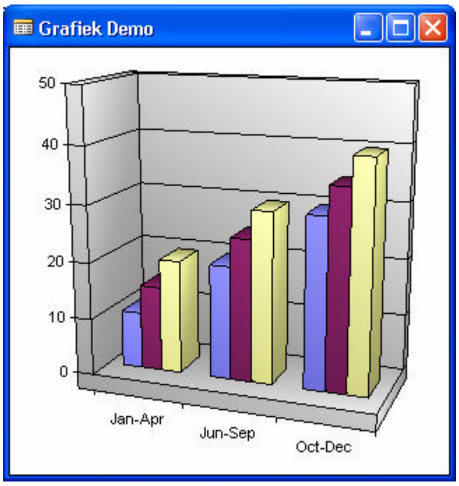


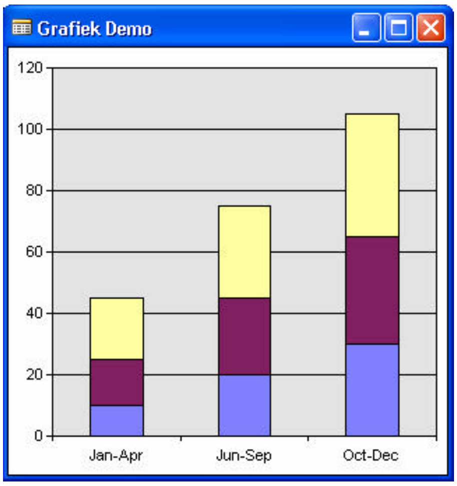
18 Column Clustered



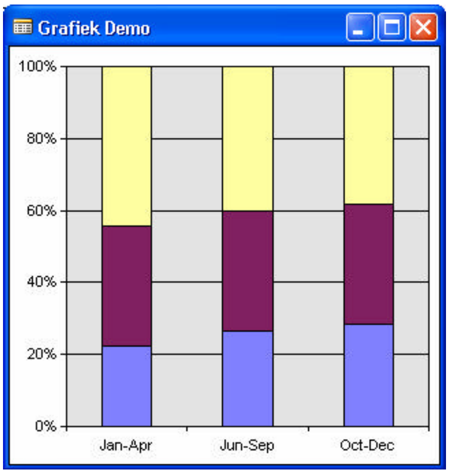
19 Column Clustered 3D

20 Column Stacked

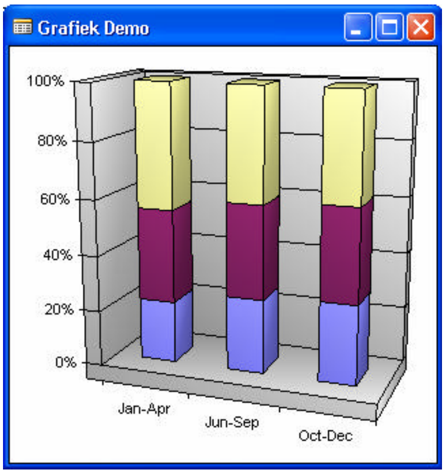




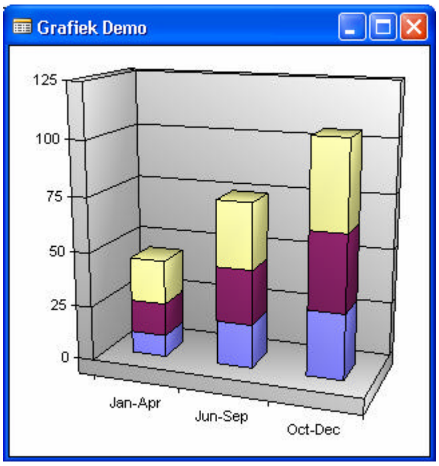
21 Column Stacked 100



22 Column Stacked 100 3D



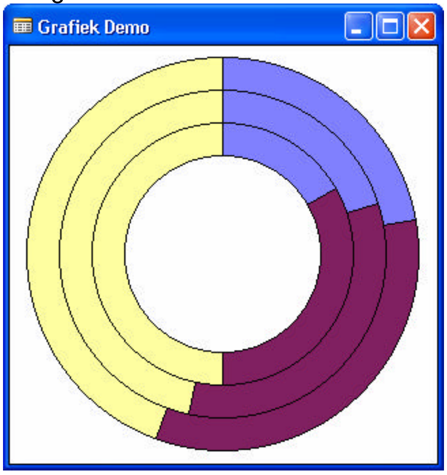
23 Column Stacked 3D



24 Combo
Sorry, this chart type is not implemented.

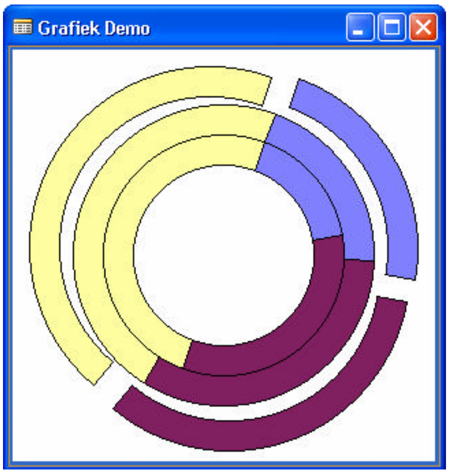
25 Combo 3D
Sorry, this chart type is not implemented.

26 Doughnut

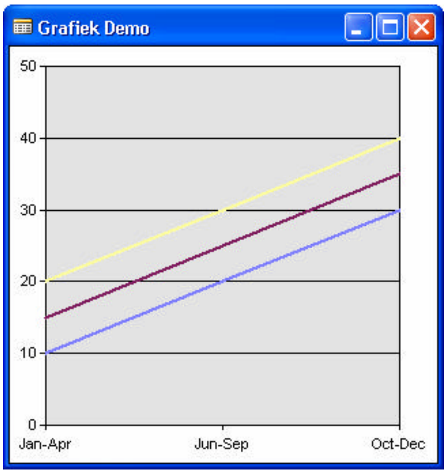


27 Doughnut Exploded

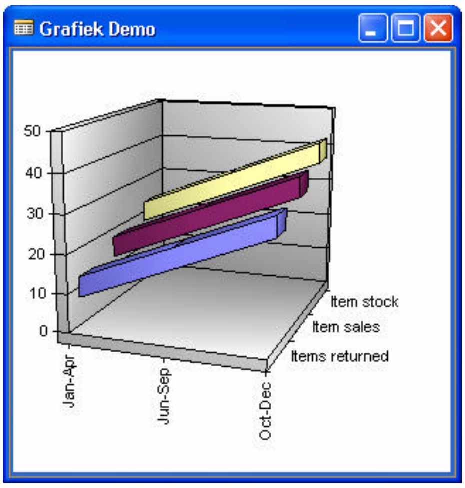
28 Line



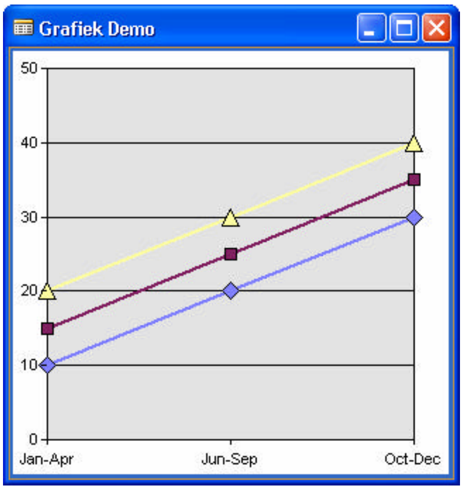
29 Line 3D



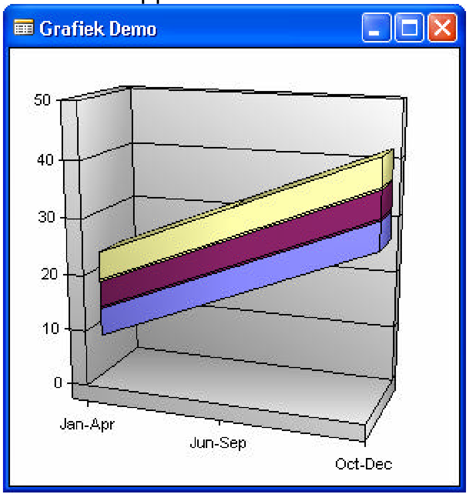
30 Line Markers



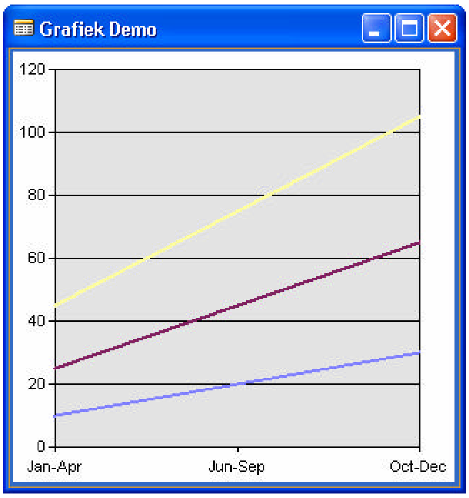
31 Line Overlapped 3D



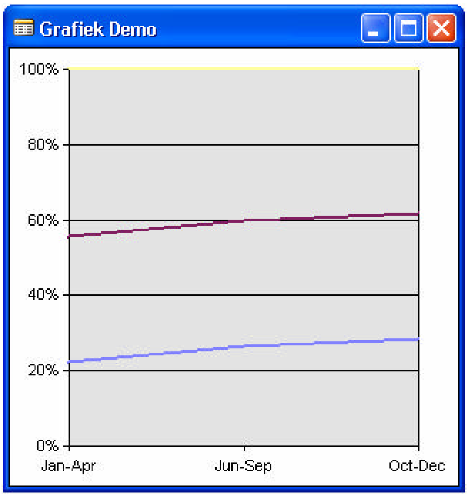
32 Line Stacked



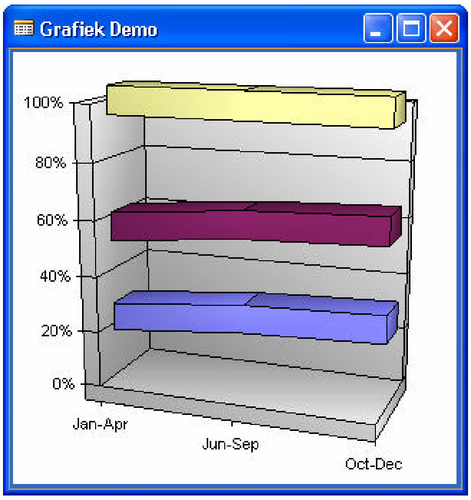
33 Line Stacked 100



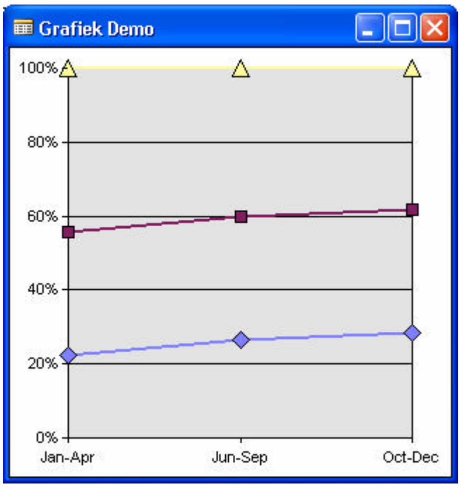
34 Line Stacked 100 3D



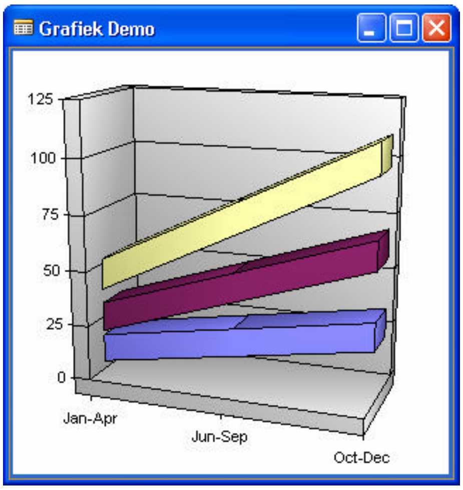
35 Line Stacked 100 Markers



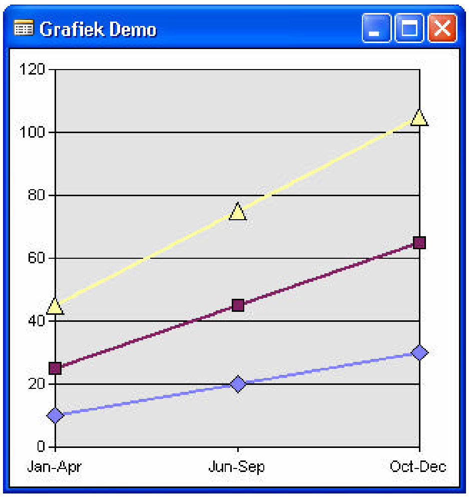
36 Line Stacked 3D



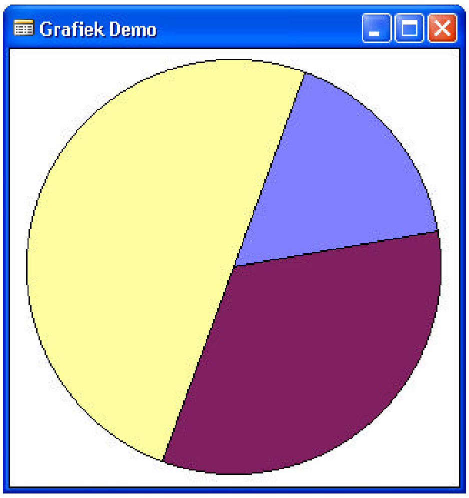
37 Line Stacked Markers



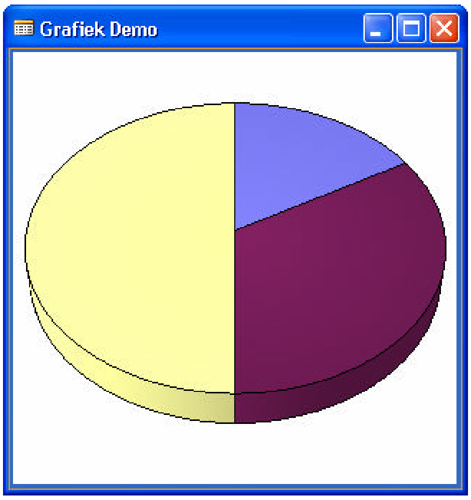
38 Pie



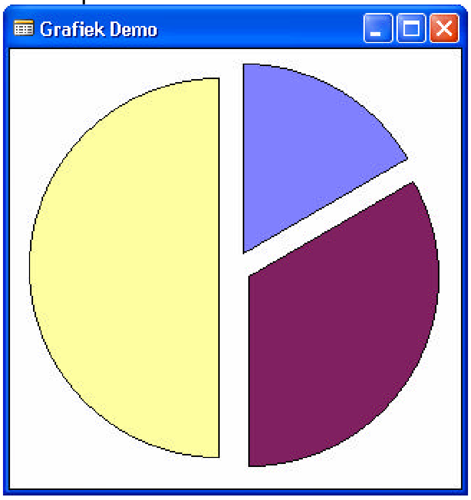
39 Pie 3D



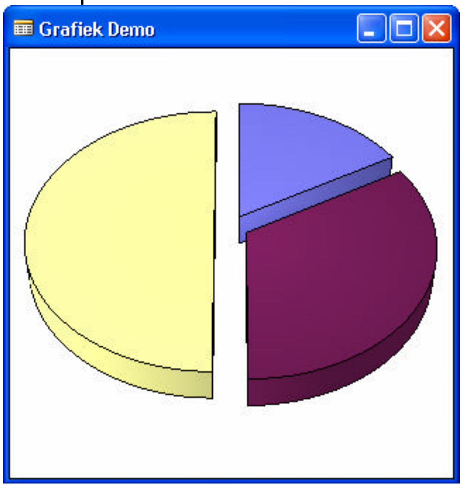
40 Pie Exploded



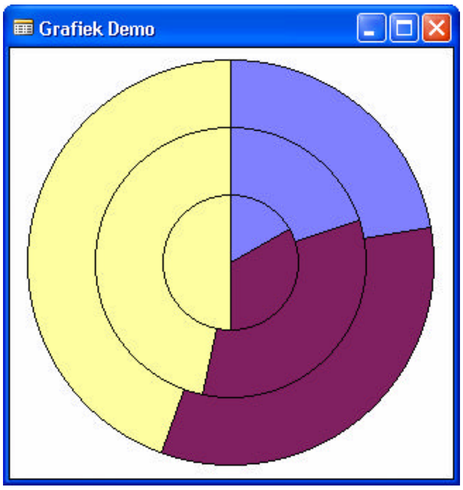
41 Pie Exploded 3D



42 Pie Stacked



43 Polar Line
Sorry, this chart type is not implemented.



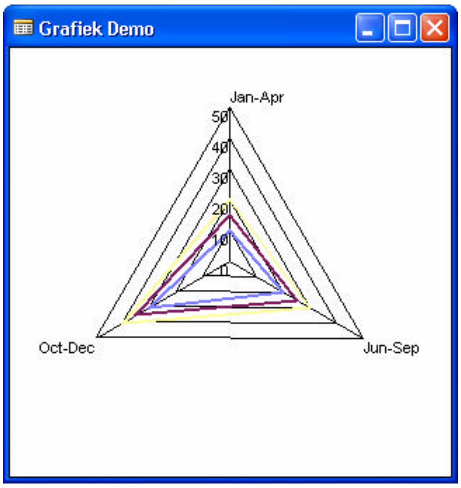
44 Polar Line Markers
Sorry, this chart type is not implemented.

45 Polar Markers
Sorry, this chart type is not implemented.

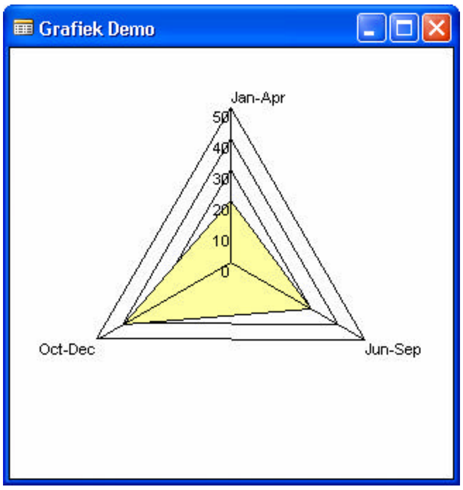
46 Polar Smooth Line
Sorry, this chart type is not implemented.

47 Polar Smooth Line Markers
Sorry, this chart type is not implemented.

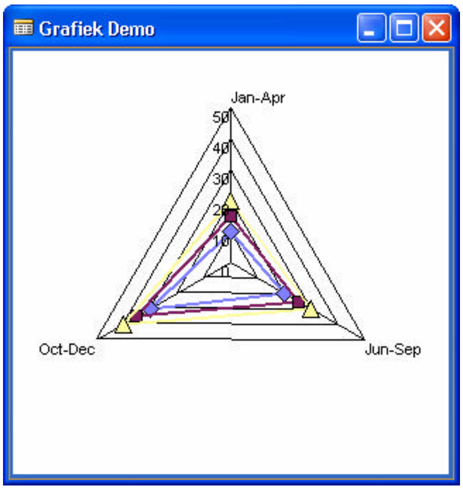
48 Radar Line



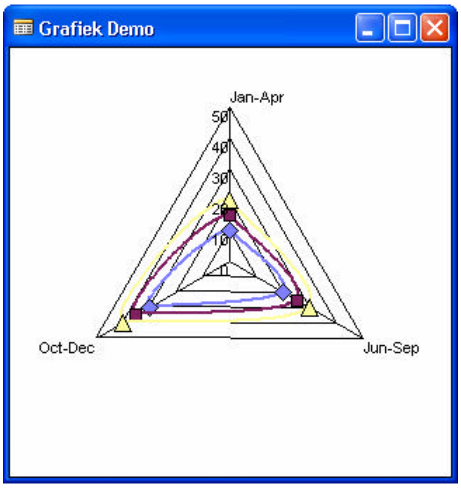
49 Radar Line Filled



50 Radar Line Markers



51 Radar Smooth Line Markers



52 Scatter Line
Sorry, this chart type is not implemented.

53 Scatter Line Filled
Sorry, this chart type is not implemented.

54 Scatter Line Markers
Sorry, this chart type is not implemented.

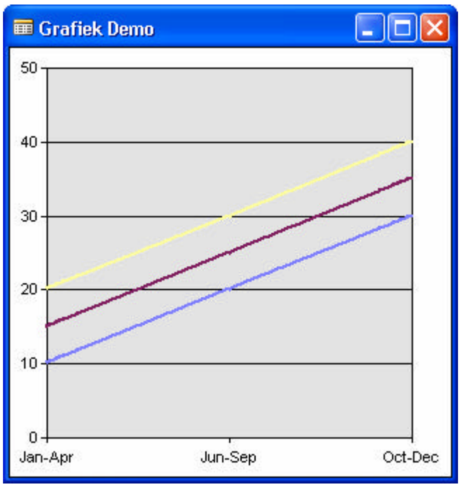
55 Scatter Markers
Sorry, this chart type is not implemented.

56 Scatter Smooth Line
Sorry, this chart type is not implemented.

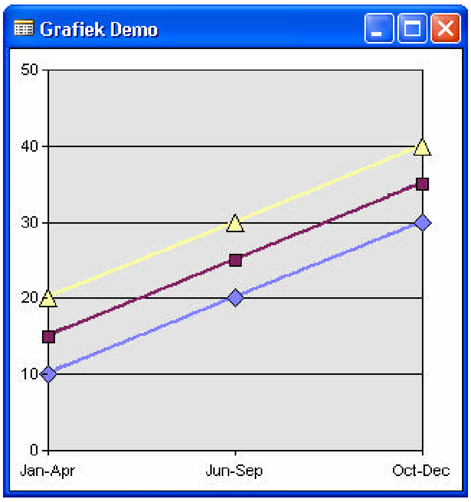
57 Scatter Smooth Line Markers

58 Smooth Line

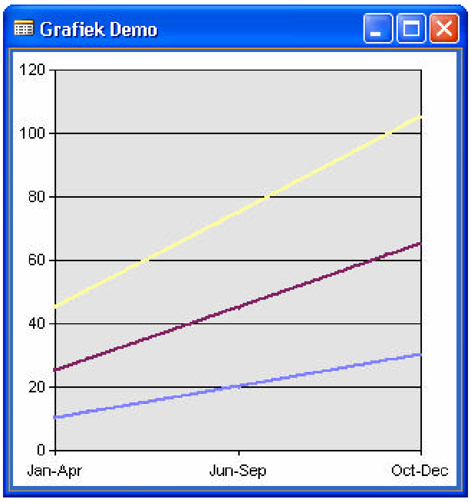
Sorry, this chart type is not implemented.



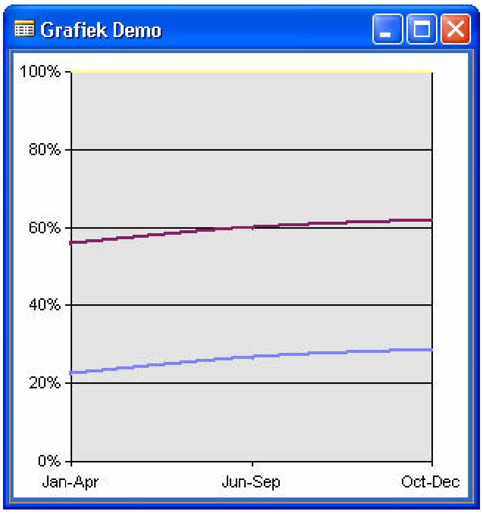
59 Smooth Line Markers



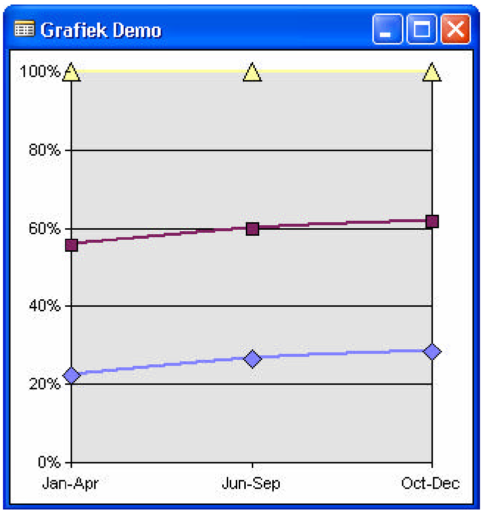
60 Smooth Line stacked



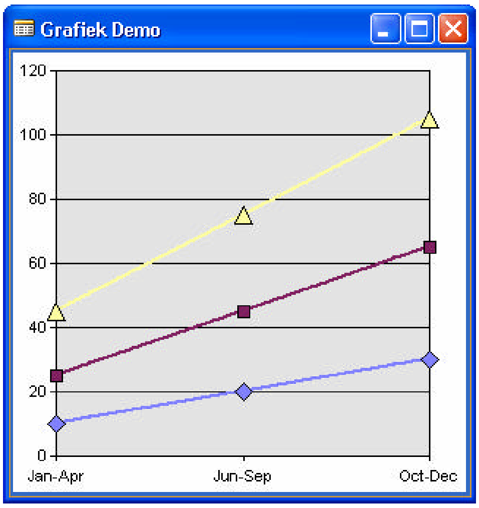
61 Smooth Line stacked 100



62 Smooth Line stacked 100 Markers



63 Smooth Line Stacked Markers



64 Stock HLC

Sorry, this chart type is not implemented.

65 Stoch OHLC

Sorry, this chart type is not implemented.

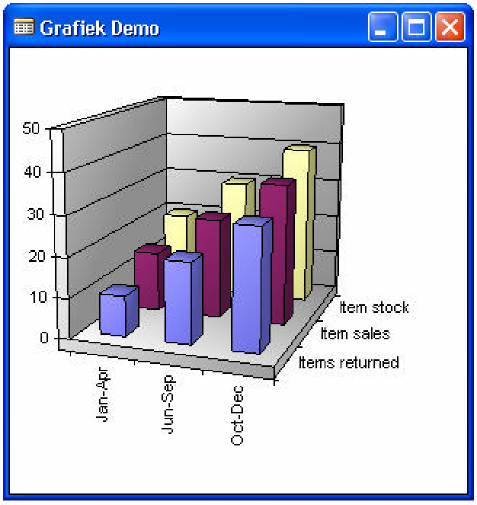
Property Let ProjectionMode <Integer>

3D Graph types can be viewed in 2 different projection modes. This can be selected with the ProjectionMode property.

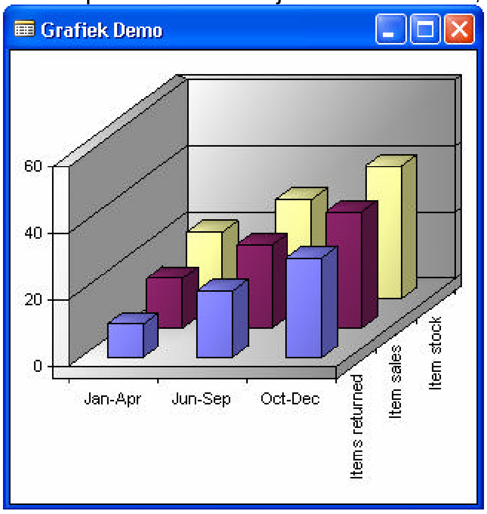
ProjectionMode 0 will result in a full 3D graph that can also be rotated. ProjectionMode 1 will always show a 3D graph from the same angle.

Example

ChartSpaceControl.ProjectionMode := 0;



ChartSpaceControl.ProjectionMode := 1;

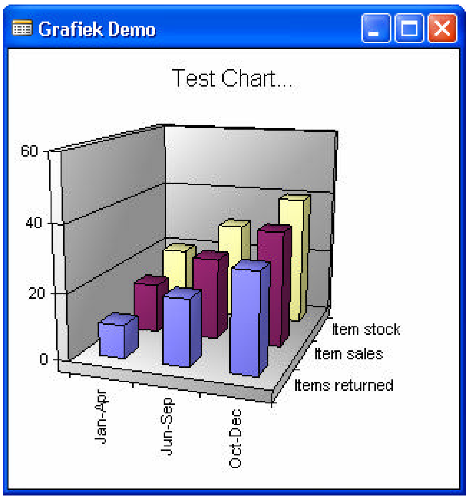


Property Title <Text>

The property Title will add a title to your chart.

Example:

`ChartSpaceControl.Title := ‘Test Chart…’`

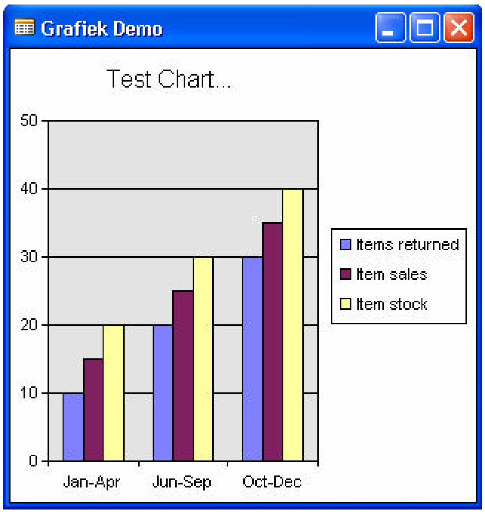


Property hasLegend <Boolean>

The hasLegend property if set to TRUE adds a legend to the chart. This is convenient for all 2D charts and some 3D graphs. Most 3D charts however shows legend items in the depth axis so there is no need to display a legend. But it can be done.

Example

`ChartSpaceControl.hasLegend := TRUE;`



Subroutine ExportPicture

Filename <text>
Width <integer>
Height <integer>

The ExportPicture subroutine can be used to save a chart to a “.bmp”, “.gif” or “.jpg” file. After exporting the picture it can be loaded into a BLOB and used in a (Navision) report.

The Width and Height parameters can be used to enter the size of the output chart. If you want it to fit in a PictureBox you can use the width and height property of the PictureBox to enter as parameters with the export subroutine. In reports you can not request the width and height of a PictureBox with C/AL code. You should hard code it into your C/AL code.

If you want to remove the picture file after you have imported the file into a BLOB use the subroutine RemovePicture. Once it is in the blob Navision does not require it to stay on the disk.

SubRoutine DeletePicture

Filename <Text>

This function can be used to delete pictures that are exported with the ExportPicture subroutine. In fact you can delete any file with this function.

When a picture is exported with ExportPicture and imported into a BLOB usually the picture file is not needed anymore. Then use DeletePicture to delete this temporary file from the disk.

SubRoutine SetTextured

Texture <String>
Format <Integer>
Placement <Integer>

With the subroutine SetTextured you can add textures to walls of the chart. There are a number of predefined textures which can be called by a number. You can also use your own texture (such as company logo for demo’s) by providing a filename instead of a number.

The predefined textures can be called by give a number in a text variable. These are the predefined textures:

0	Blue Tissue Paper	1	Bouquet
2	Brown Marble	3	Canvas
4	Cork	5	Denim
6	Fish Fossil	7	Granite
8	Green Marble	9	Medium Wood
10	News Print	11	Oak
12	Paper Bag	13	Papyrus
14	Parchment	15	Pink Tissue Paper
16	Purple Mesh	17	Recycled Paper
18	Sand	19	Stationary
20	Walnut	21	Water Droplets
22	White Marble	23	Woven Mat

With the format parameter you can specify how the texture is displayed.

0	Tile	1	Stack
3	Stack Scale	4	Stretch
5	Stretch Plot		

With the placement parameter you can specify where the texture is displayed.

0	All Faces	1	End
2	End Sides	3	Front
4	Front End	5	Front Sides
6	Project Front	7	Sides

Remember that some combinations of parameters are not possible and produce errors.

Property AspectRatio <integer>

With the AspectRation property you can ‘stretch’ the chart. The values that are allowed are 0–500. This property only affects 3D charts.

Property ThickNess <Integer>

With the ThickNess property you can set the thickness of 3D line charts. The values that are allowed are 0-100.

Property Depth <Integer>

With the depth property you can set the depth of an 3D chart. The values that are allowed are 0-500.

Property AmbientLightIntensity <Double>

With the AmbientLightIntensity you can make the 3D chart lighter or darker. The values that are allowed are 0-1. This property requires a decimal value.

Property LightRotation <Integer>

With the LightRotation property you can set the angle where the of the light source of an 3D chart. The values that are allowed are 0-360.

Property Angle <Integer>

With the Angle property you can set the angle of an 3D chart. The values that are allowed are 0-360.

This property only has effect when the ProjectionMode is set to 1.

Property Inclination <Integer>

With the Inclination property you can set the horizontal inclination of an 3D chart. The values that are allowed are -90-90.

This property only has effect when the ProjectionMode is set to 0.

Property LightNormal <Double>

With the LightNormal property you can set the light direction deviation of an 3D chart. The values that are allowed are 0-1. This property requires a decimal value.

Property Rotation <Integer>

With the property Rotation you can rotate the 3D chart. The values that are allowed are 0-360.

This property only has effect when the ProjectionMode is set to 0.

Property GapDepth <Integer>

The property GapDepth sets the size of the gap at the depth axis of an 3D chart. The values that are allowed are 0-500.

Property Perspective <Integer>

The Property Perspective sets the perspective of an 3D chart. The values that are allowed are 0-80.

This property only has effect when the ProjectionMode is set to 0.

Property DirectionalLightInclination <Integer>

With the DirectionalLightInclination property you can set the vertical inclination of an 3D chart. The values that are allowed are -90-90.

Property DirectionalLightIntensity <Double>

With the DirectionalLightIntensity property you can set the vertical light intensity of an 3D chart. The values that are allowed are 0-1. This property requires a decimal value.