THREE D GRAPHICS

Preliminary Draft

(Chart Enhancement Macros for Business Objects Web Intelligence 3.x)

FEBRUARY, 2010

Three |D| Graphics | 11340 West Olympic Blvd., #352 | Los Angeles, CA. 90064

Telephone: 1.310.231.3330 | Fax: 1.310.231.3303

Web: http://www.threedgraphics.com

NOTICES:

Copyright 2010 by Three |D| Graphics. All rights reserved. This document may not be reproduced or disclosed in whole or in part by any means without the written consent of Three |D| Graphics.

Three |D| Graphics

11340 West Olympic Blvd., #352, Los Angeles, CA. 90064

Telephone: 1.310.231.3330

Fax: 1.310.231.3303

Web: http://www.threedgraphics.com

FEBRUARY, 2010

This document can be found on-line at:

http://threedgraphics.com/tdg/products/tools/bochart/html_docs/bochart.htm

Table of Contents

Section 1: Overview

WebI:Chart is a replacement charting library for Business Objects Web Intelligence 3.x. It adds significant new charting capabilities for Business Objects designers and developers. This “replacement library” is 100% compatible with your Business Objects Web Intelligence 3.x product. It simply expands your charting and graphing capabilities. Features:

- Box Plots
- 3D Scatter Charts
- User programmable lines on any axis (or free floating)
- User programmable markers at any point on the chart
- Error Bars
- Pivot functions (Swap series/group, data reversal, etc...)
- Conditional Coloring and Formatting
- Drop Shadow and Alpha Channel transparency effects
- High-Quality Rendering
- Expanded Gauge Functionality
- Advanced Pie Label Layout Engine

Access to these enhanced charting features is provided through a set of special Macro commands that are added to your custom chart module.

Setup Instructions

1) Find the exiting chart module (hs_pgsdk32.dll) for your Business Objects Enterprise installation. By default, it should be at the following location: C:\Program Files\business objects\businessobjects enterprise 12.0\win32_x86\

2) MAKE A BACKUP COPY of the existing chart module: hs_pgsdk32.dll.

3) Replace the chart module with the hs_pgsdk32.dll included in the WebI:Chart zip package.

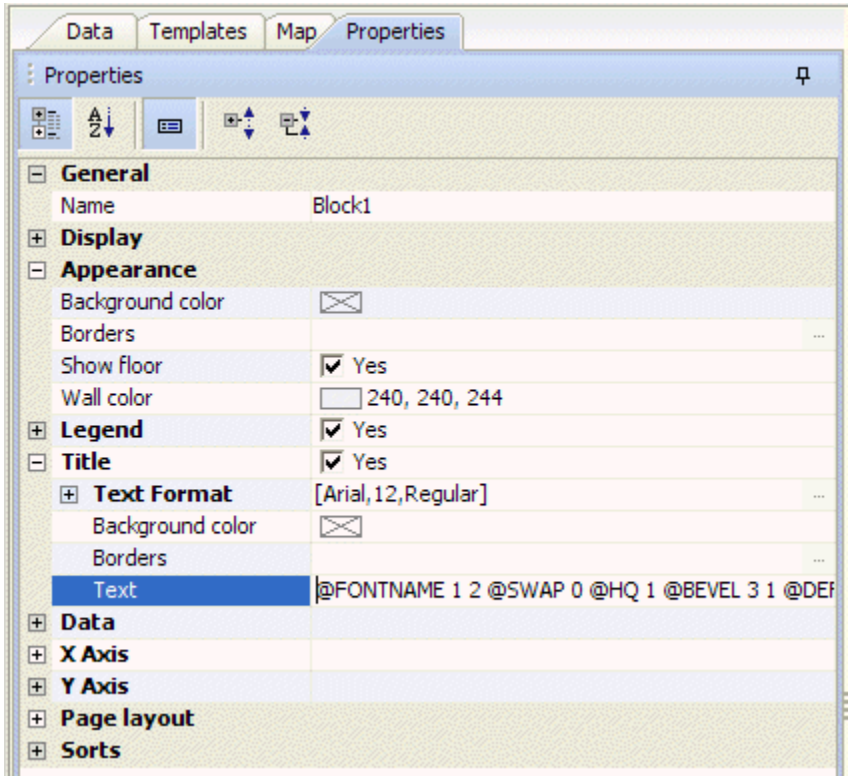
4) Copy the cairo.dll in the WebI:Chart zip package to the same folder.

5) Restart the Business Objects Intelligence Server, usually through the Business Objects Central Configuration Manager.

NOTE: This is a demo version of WebI:Chart. It is set to expire March 1, 2010. After that time, you will need to restore the original hs_pgsdk32.dll to restore the original functionality.

General Usage Notes

WebI:Chart macros can be inserted in the chart title or axis label text fields in the chart’s properties panel.



In the Properties panels, these fields are located at:

- Title/Text Format/Text
- X-Axis/Label/Other Label
- Y-Axis/Label/Other Label
- Z-Axis/Label/Other Label

If you need to use one of these files to define a chart title or label, append a tilde (~) character and a space to the title and add the macro after the space. The space after the tilde is important. If it is not included, the macro will be ignored.

EXAMPLE:

```
TitleText~ @3DSCAT
```

WebI:Chart macros are not case-sensitive. All macros begin with an at-sign (@). There should be no space between the at-sign and the macro (i.e., @SWAP, not @ SWAP). Most macros include one or more parameters that further define the action of the macro. There must be one space after the macro and before the first parameter and a space between each subsequent parameter.

EXAMPLE:

```
@USER_SERIES 2 3.6 3.8 MyNewSeries
```

Use a space to separate multiple macros.

EXAMPLE:

```
@BP2 @MK 8
```

If the macro includes a string and multiple macros are defined in the title field, append a tilde (~) character to the end of the string parameter to indicate the end of the first macro and beginning of the next.

EXAMPLE:

```
@AGL 2 Alias Label~ @BP2
```

For macros that can be applied to a particular series in a chart, the *nSeries* parameter defines the series to which the macro is applied. In most cases, the *nSeries* parameter can be assigned a value in the range: minus one (-1)...*n* (where: *n* is the total number of series in the chart). For these macros, minus one is a special value that will apply the macro to all series in the chart. A value of zero (0) selects series 1.

Order of Evaluation

Because chart enhancement macros can be defined in chart title and label fields, you could potentially define conflicting macros. To avoid this potential conflict, the macros are evaluated in the following order:

1. Z-Axis Label (Scatter Charts Only)
2. Y-Axis Label
3. X-Axis Label
4. Title

Multiple macros in the same title or label field are parsed from left-to-right. For example, assume the following macros are defined in the Title field:

```
@SWAP 0 @SWAP 1
```

In this example, "@SWAP 1" would be used because it is defined AFTER "@SWAP 0".

Persistence

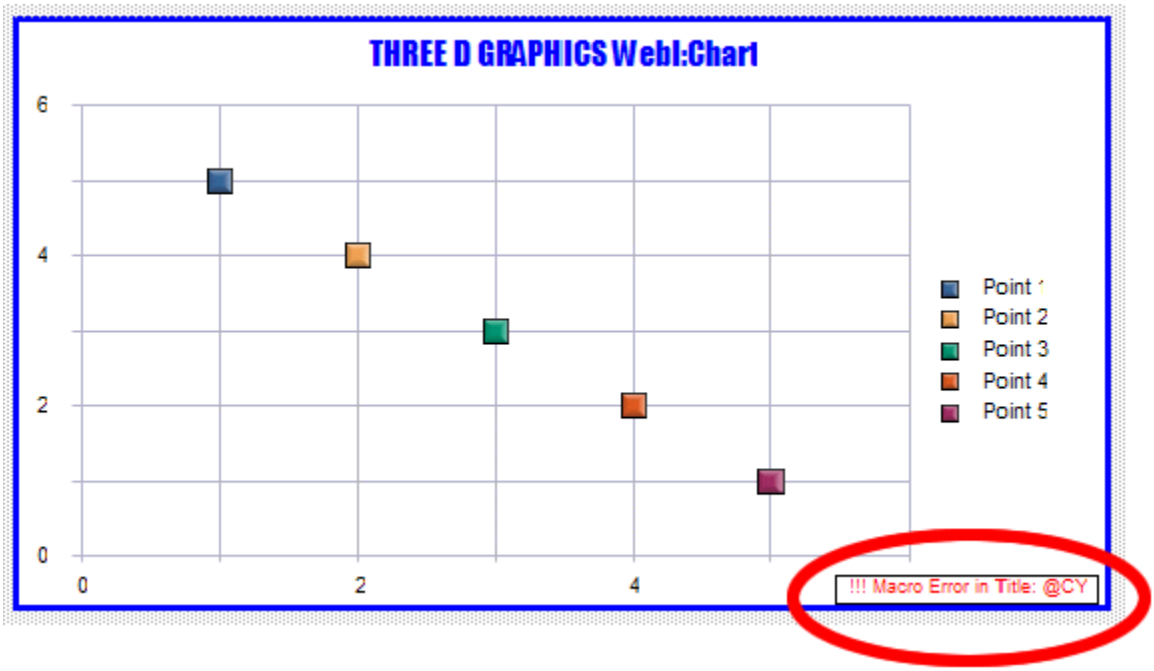
Many of the chart enhancement macros set a property of the chart and that property will remain set, even if the macro is removed from the field. For example, if "@SWAP 1" is used to reverse the series/groups in the chart, the series/groups will remain reversed even if the "@SWAP 1" macro is removed. A "@SWAP 0" macro is needed to return the series/groups to their normal/default order.

Some of the chart enhancement macros are NOT persistent and their effect will disappear when the macro is removed. For example the user-defined lines (set by @X, @Y, and @XY) and user-defined series labels (set by @ASL) only remain in the chart while the macro definition is in place. When the macro is removed, the user-defined line and/or series labels disappear.

The description of each macro identifies whether or not the macro is persistent.

Error Reporting

If your macro encounters an error, a red message will be displayed in the lower right corner of the chart.



The most common errors are:

1) missing parameter(s)

2) parameter is assigned an out-of-range value

3) missing space between macro and parameter or consecutive parameters.

Section 2: Auto Arrange Macros

The following macros can be used to automatically arrange objects in a chart for optimal placement and appearance:

- @AA; Automatically arrange objects in a chart.
- @AA2; Automatically arrange objects in a chart with Frame adjustment.

@AA (Auto Arrange)

This macro automatically arranges elements in a chart.

SYNTAX:

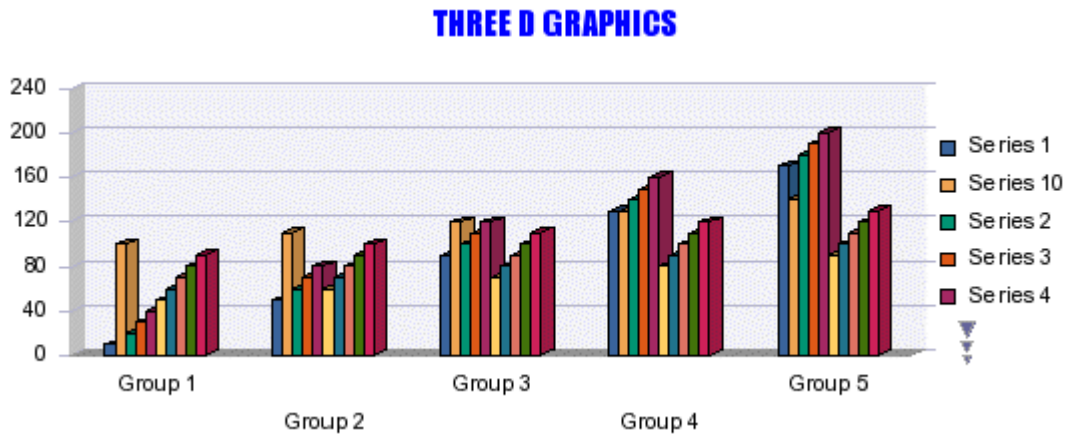
```
@AA
```

PARAMETERS:

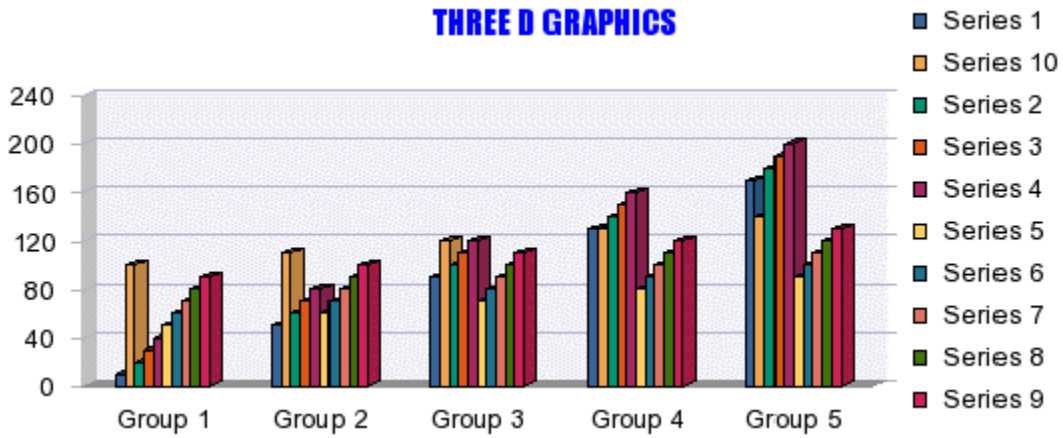
None

EXAMPLE:

```
Before @AA
```




```
@AA
```



PERSISTENT:

YES

@AA2 (Auto Arrange with Frame Adjustment)

This macro automatically arranges elements in a chart. It is the same as the @AA macro with the following exceptions:

1) It does not auto-size font labels. Each label keeps its current font size.

2) The *nFixup* parameter can be used to move the bottom of the chart frame up or down to provide more or less space for the X-axis labels

SYNTAX:

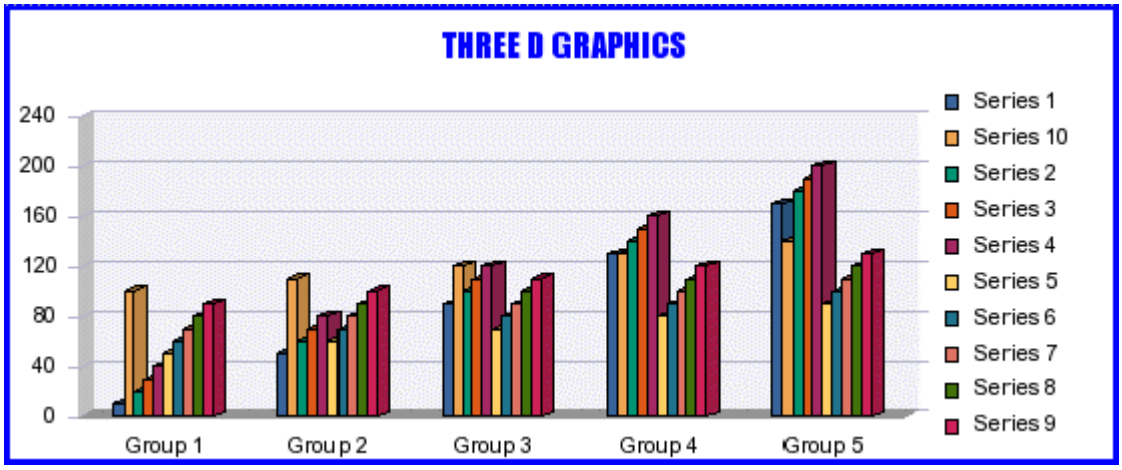
```
@AA2 nFixup
```

PARAMETERS:

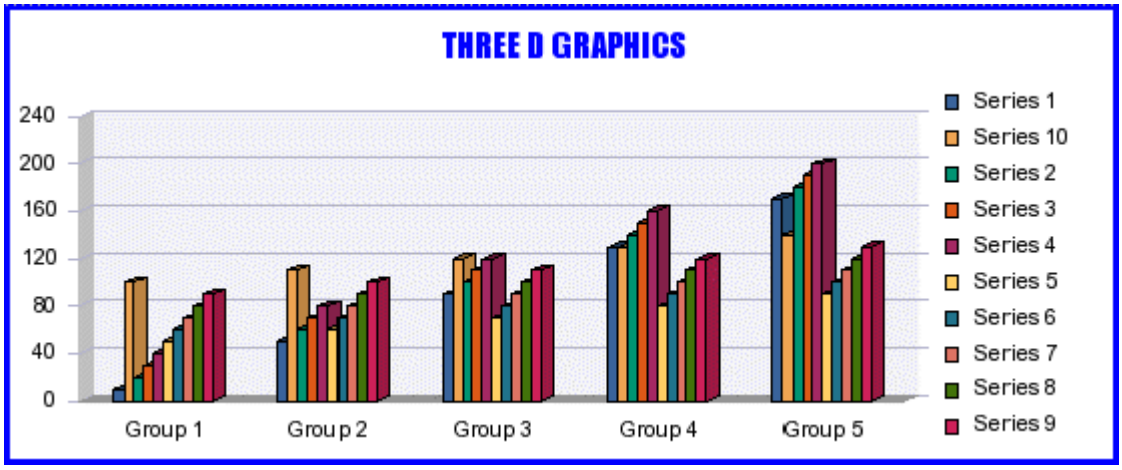
nFixup; -16000...16000 specifies how much to move the bottom of the chart frame up or down.

EXAMPLE:

```
THREE D GRAPHICS~ @FONTNAME 1 2 @AA2 100
```




```
THREE D GRAPHICS~ @FONTNAME 1 2 @AA2 -1000
```



PERSISTENT:

NO


WebI: CHART

Section 3: Axis Macros

These macros can be used to control the appearance of the chart's axes, gridlines, and scales:

- @AXIS; Assign a Series to an Axis
- @DX; X-Axis Divisions
- @DY; Y-Axis Divisions
- @DY2; Y2-Axis Divisions
- @GRIDLINES_ON_TOP; Draw gridlines in front of/behind risers
- @GX; X-Axis Grid Style
- @GY; Y-Axis Grid Style
- @SC; Y-Axis Scale
- @SCALE_INTERVAL; Scale Interval on the Y1, Y2, or X-Axis
- @SCX; X-Axis Scale
- @SCY2; Y2-Axis Scale
- @Y1_INVERT; Y1-Axis Invert
- @Y1BASE; Y1-Axis Base Line
- @Y2_INVERT; Y2-Axis Invert
- @Y2BASE; Y2-Axis Base Line

@AXIS (Assign Series to Axis)

In dual-Y and bi-polar axis charts, this macro assigns a series to the Y1 or Y2 axis.

SYNTAX:

```
@AXIS nSeries bAxis
```

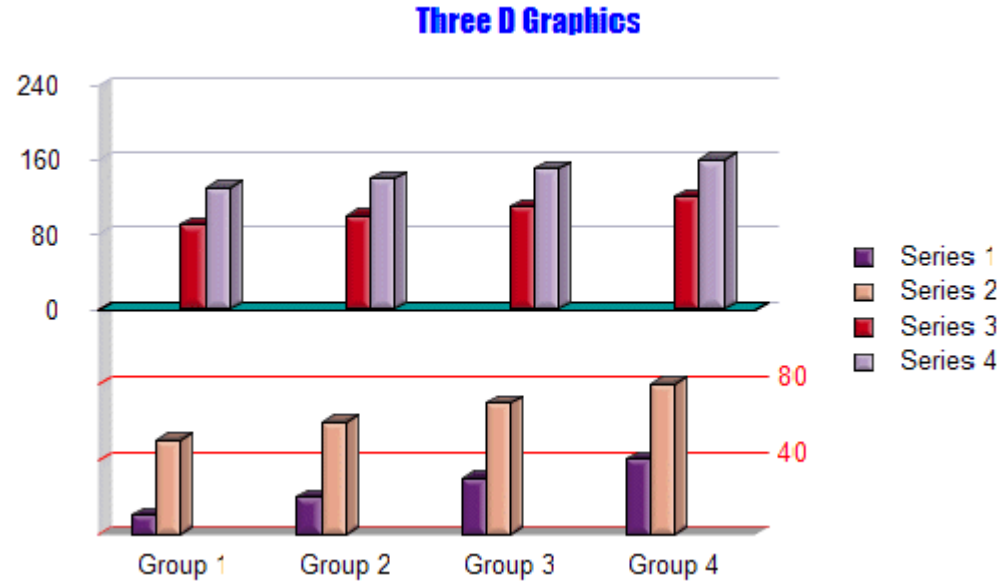
PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

bAxis; 0/1
0=assign *nSeries* to Y1-Axis
1=assign *nSeries* to Y2-Axis

EXAMPLE:

```
@GRAPHTYPE 18
@AXIS 0 1
@AXIS 1 1
@AXIS 2 0
@AXIS 3 0
```



PERSISTENT:

YES

@DX (X-Axis Divisions)

This macro sets the number of division on the X-Axis. It can only be used in a chart with a true X-Axis (e.g., Scatter, Bubble, or Polar).

SYNTAX:

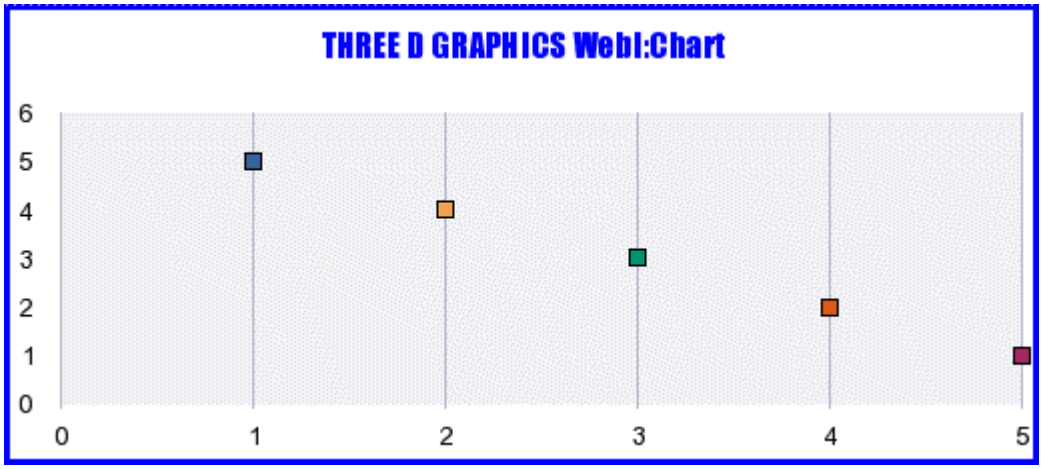
```
@DX nDivisions
```

PARAMETERS:

nDivisions; 1...99 divisions on the X-Axis

EXAMPLE:

```
@DX 4
```



PERSISTENT:

YES

ALSO SEE:

@DY to set the number of divisions on the Y-Axis

@DY2 to set the number of divisions on the Y2-Axis

@DY (Y-Axis Divisions)

This macro sets the number of division on the Y-Axis.

SYNTAX:

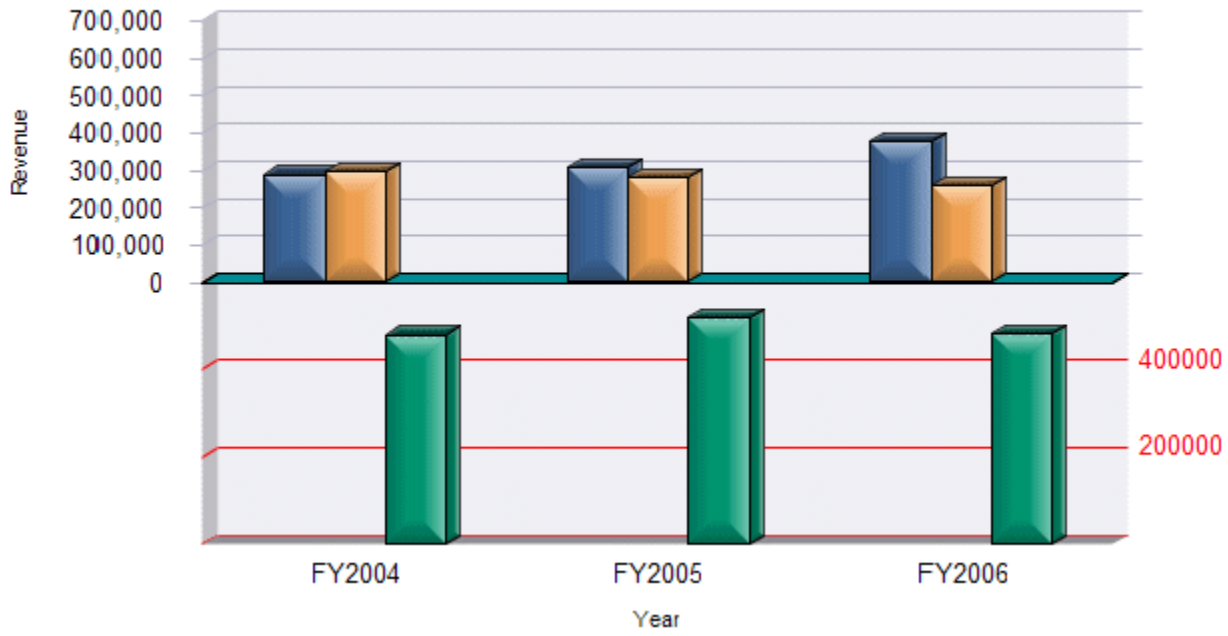
```
@DY nDivisions
```

PARAMETERS:

nDivisions; 1...99 divisions on the Y-Axis

EXAMPLE:

```
@DY 6
```



PERSISTENT:

YES

ALSO SEE:

@DX to set the number of divisions on the X-Axis

@DY2 to set the number of divisions on the Y2-Axis

@DY2 (Y2-Axis Divisions)

This macro sets the number of division on the Y2-Axis.

SYNTAX:

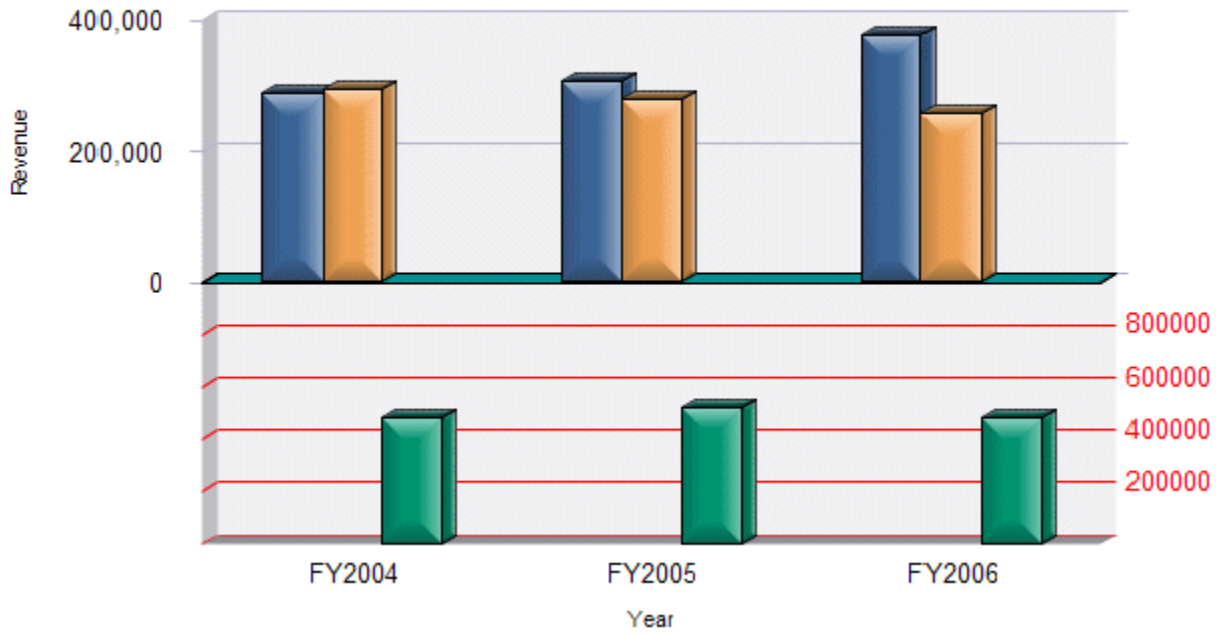
```
@DY2 nDivisions
```

PARAMETERS:

nDivisions; 1...99 divisions on the Y2-Axis

EXAMPLE:

```
@GRAPHTYPE 18
@AXIS 1 1
@DY 2
@DY2 4
```



PERSISTENT:

YES

ALSO SEE:

@DX to set the number of divisions on the X-Axis

@DY to set the number of divisions on the Y-Axis

@GRIDLINES_ON_TOP

This macro draws gridlines behind or in front of the chart's risers.

SYNTAX:

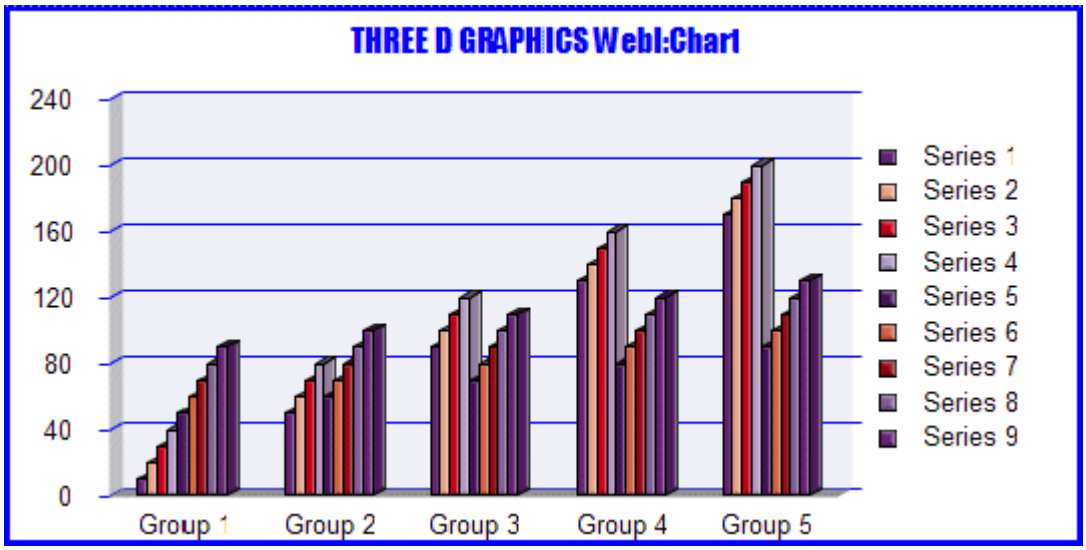
```
@GRIDLINES_ON_TOP bShowGridlinesOnTop
```

PARAMETERS:

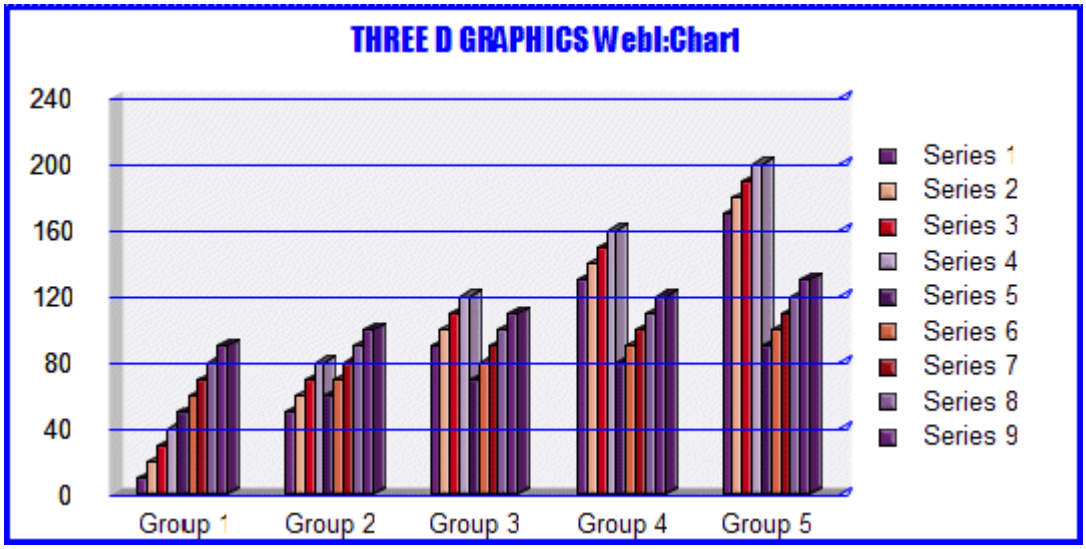
bShowGridlinesOnTop; 0 = Draw gridlines behind risers, 1 = Draw gridlines in front of risers.

EXAMPLE:

```
@GRIDLINES_ON_TOP 0
```




```
@GRIDLINES_ON_TOP 1
```



@GX (X-Axis Grid Style)

In a chart with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), this macro sets the grid style to use on the X-Axis. In other chart types, it sets the grid style to use on the group/O1-Axis.

SYNTAX:

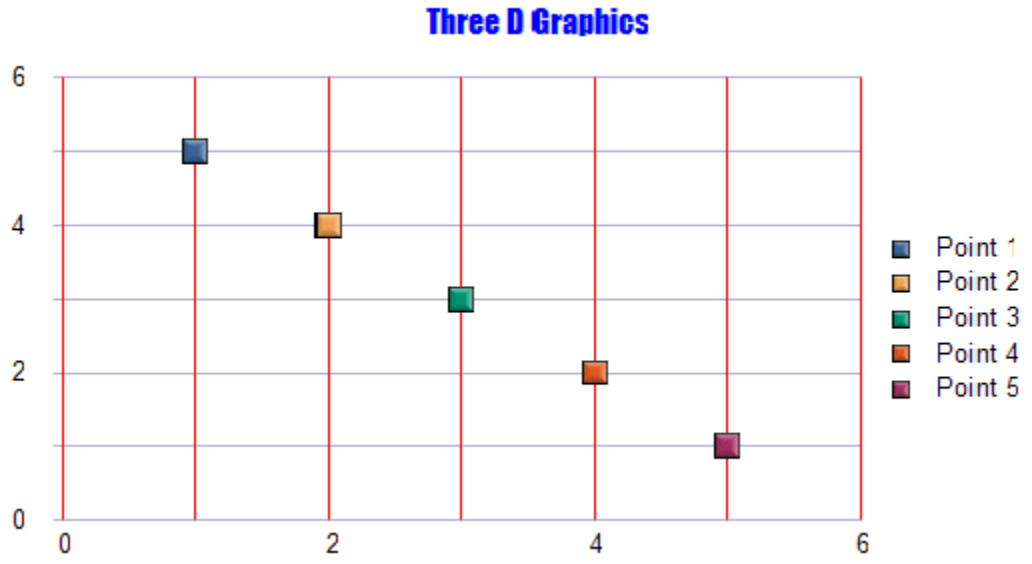
```
@GX nStyle
```

PARAMETERS:

nStyle; 0...5 selects one of the following grid/tick styles:
0 = No Grids or Ticks
1 = Standard Grid. No Tick
2 = Standard Grid. Outer Tick.
3 = No Grid. Inner Tick.
4 = No Grid. Outer Tick.
5 = No Grid. Inner and Outer Tick.

EXAMPLE:

```
@GX 1
```



PERSISTENT:

NO

ALSO SEE:

@GY

@GY (Y-Axis Grid Style)

This macro sets the Grid/Tick style on the Y-Axis.

SYNTAX:

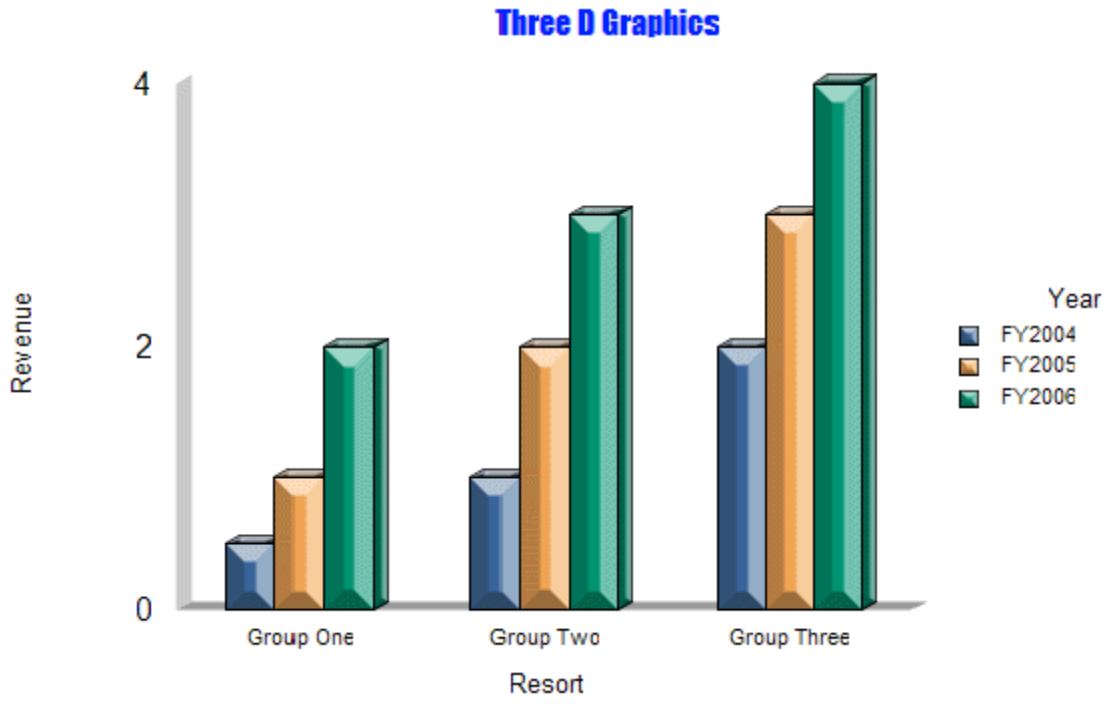
```
@GY nStyle
```

PARAMETERS:

nStyle; 0...5 selects one of the following grid/tick styles:
0 = No Grids or Ticks
1 = Standard Grid. No Tick
2 = Standard Grid. Outer Tick.
3 = No Grid. Inner Tick.
4 = No Grid. Outer Tick.
5 = No Grid. Inner and Outer Tick.

EXAMPLE:

```
@GY 0
```



PERSISTENT:

NO

ALSO SEE:

@GX

@SC (Y-Axis Scale)

This macro sets the minimum and maximum values that can appear on the Y-axis.

SYNTAX:

```
@SC fMin fMax
```

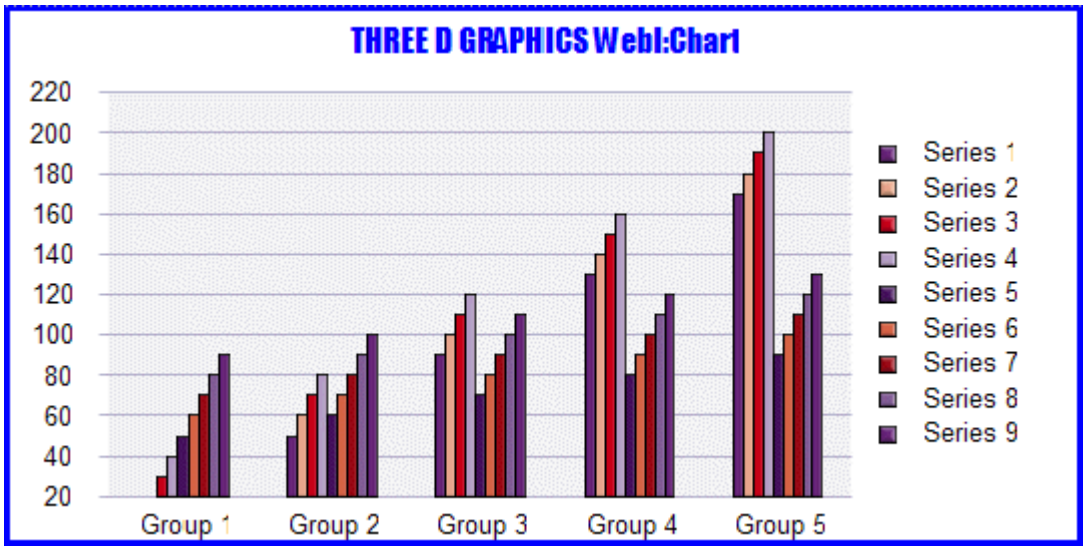
PARAMETERS:

fMin; Minimum value to show on the Y-Axis

fMax; Maximum value to show on the Y-Axis

EXAMPLE:

```
@SC 20 220
```



PERSISTENT:

YES

NOTES:

If *fMin* and *fMax* parameters are BOTH set to 0.0, the Y-Axis scale mode is reset to automatic (i.e., the charting library automatically calculates the Y-Axis scale based on the values in the data set).

ALSO SEE:

- @SCY2 to set Y2-Axis scale
- @SCX to set X-Axis scale
- @SCALE_INTERVAL; Scale Interval on the Y1, Y2, or X-Axis

@SCALE_INTERVAL (Scale Interval on Y1, Y2, or X-Axis)

This macro sets the interval to use on the Y1, Y2, or X Axis

SYNTAX:

```
@SCALE_INTERVAL nAxis fInterval
```

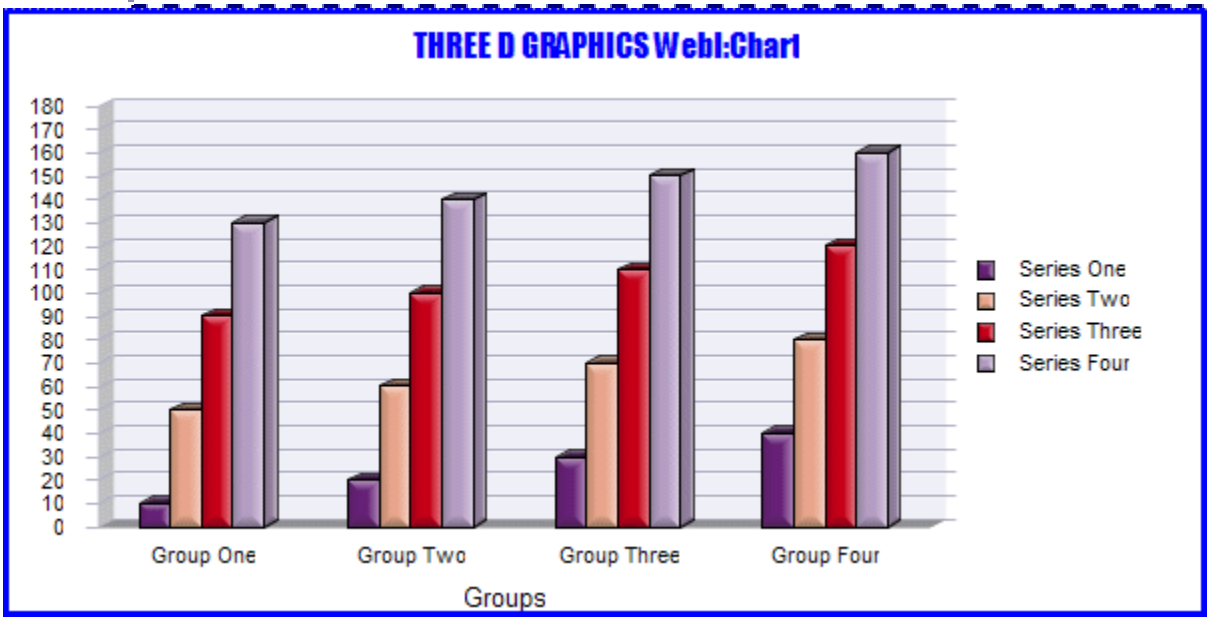
PARAMETERS:

nAxis; 0...2. 0 = Y1-Axis, 1 = Y2-Axis, 2 = X-Axis

fInterval; Scale interval value

EXAMPLE:

```
@SCALE_INTERVAL 0 10
```



ALSO SEE:

- @SCY to set Y1-Axis scale
- @SCY2 to set Y2-Axis scale
- @SCX to set X-Axis scale

@SCX (X-Axis Scale)

This macro sets the minimum and maximum values that can appear on the X-Axis. It can only be used in a chart with a true X-Axis (e.g., Scatter, Bubble, Polar, etc).

SYNTAX:

```
@SCX fMin fMax
```

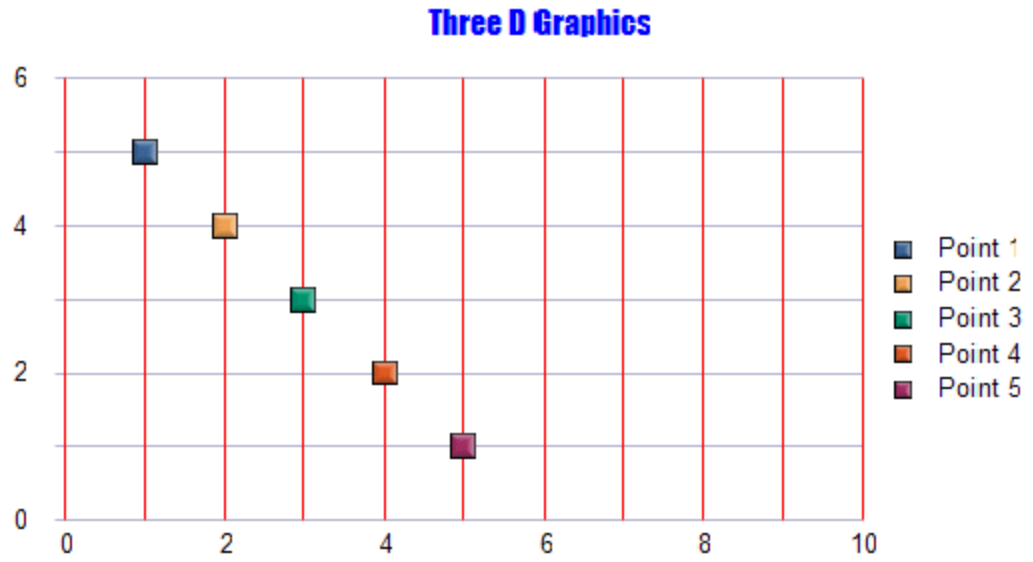
PARAMETERS:

fMin; Minimum value to show on the X-Axis

fMax; Maximum value to show on the X-Axis

EXAMPLE:

```
@SCX 0 10
```



PERSISTENT:

YES

NOTES:

If *fMin* and *fMax* parameters are BOTH set to 0.0, the X-Axis scale mode is reset to automatic (i.e., the charting library automatically calculates the X-Axis scale based on the values in the data set).

ALSO SEE:

- @SC to set Y-Axis scale
- @SCY2 to set Y2-Axis scale

@SCY2 (Y2-Axis Scale)

This macro sets the minimum and maximum values that can appear on the Y2-axis in a dual-axes chart.

SYNTAX:

```
@SCY2 fMin fMax
```

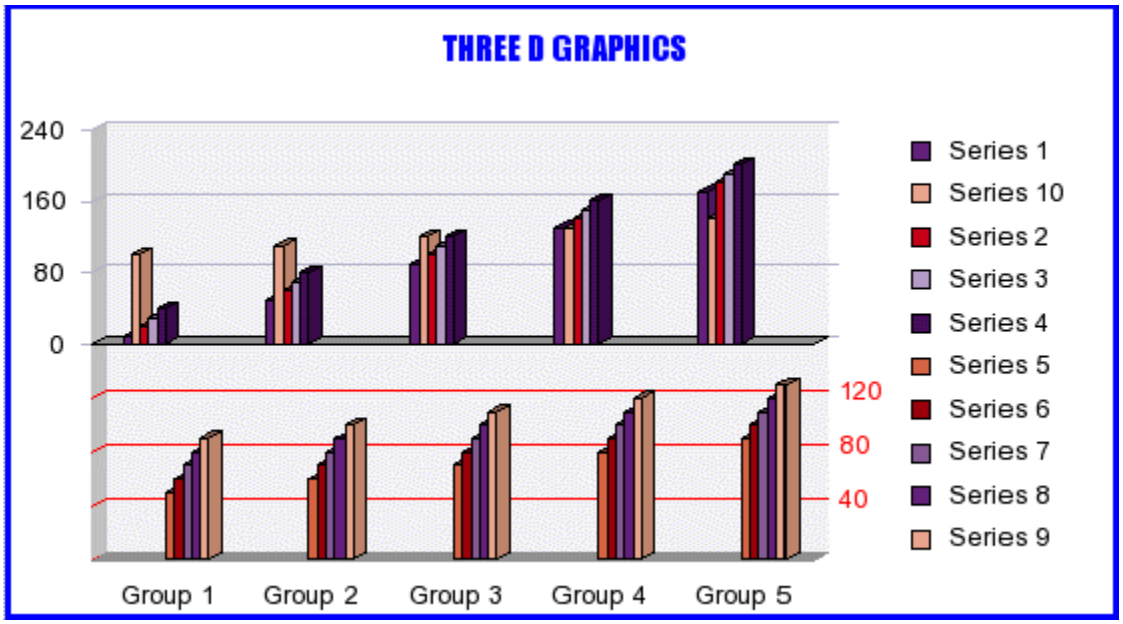
PARAMETERS:

fMin; Minimum value to show on the Y2-Axis

fMax; Maximum value to show on the Y2-Axis

EXAMPLE:

```
@SCY2 0 0
```



PERSISTENT:

YES

NOTES:

If *fMin* and *fMax* parameters are BOTH set to 0.0, the Y2-Axis scale mode is reset to automatic (i.e., the charting library automatically calculates the Y2-Axis scale based on the values in the data set).

ALSO SEE:

- @SC to set Y-Axis scale
- @SCX to set X-Axis scale

@Y1_INVERT (Y1-Axis Invert)

This macro inverts the values and labels on the Y1-axis.

SYNTAX:

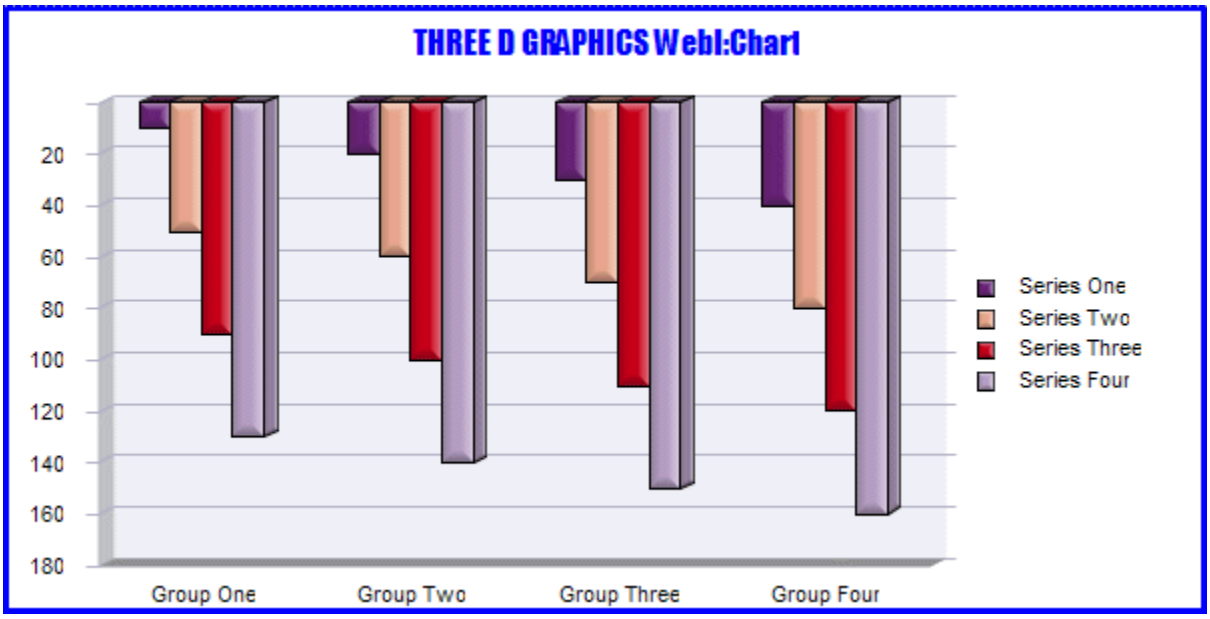
```
@Y1_INVERT bInvert
```

PARAMETERS:

bInvert; 0=Draw values and labels normally. 1=Invert values and labels.

EXAMPLE:

```
@Y1_INVERT 1
```



PERSISTENT:

NO

@Y1BASE (Y1-Axis Base Line)

This macro specifies a baseline position for the Y1-axis. The default value is 0.0. Values greater than *fBase* draw "Up" from the baseline. Values less than *fBase* draw "Down" from the baseline.

SYNTAX:

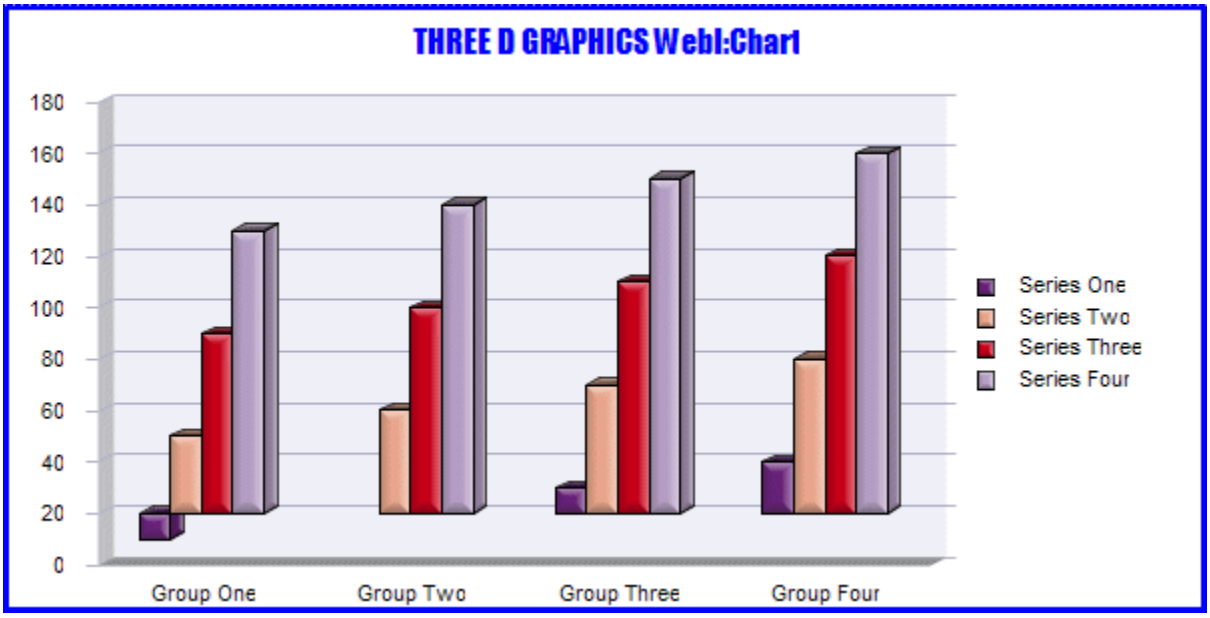
```
@Y1BASE fBase
```

PARAMETERS:

fBase; Base line value

EXAMPLE:

```
@Y1BASE 20
```



PERSISTENT:

NO

@Y2_INVERT (Y2-Axis Invert)

This macro inverts the values and labels on the Y2-axis.

SYNTAX:

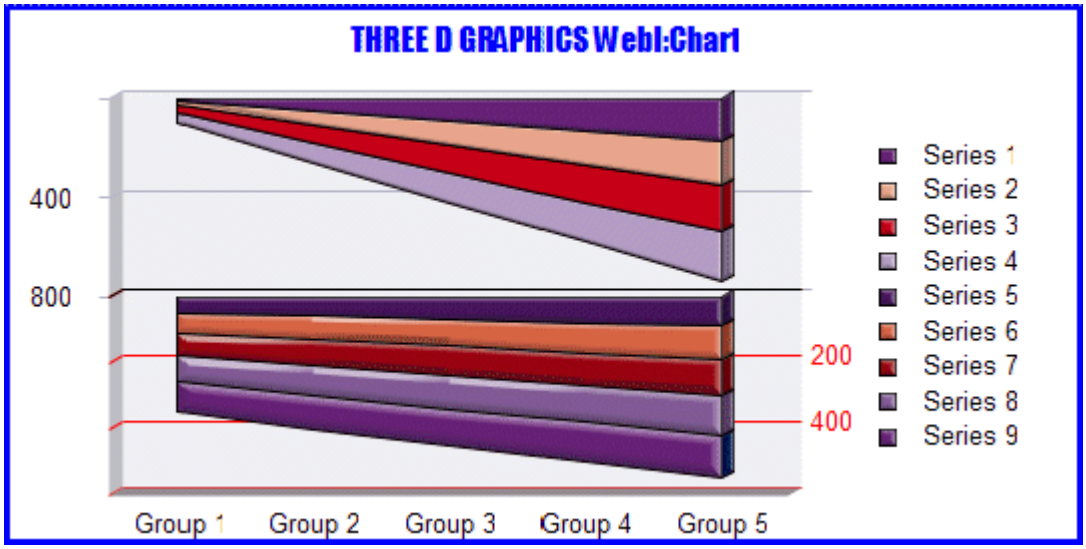
```
@Y2_INVERT bInvert
```

PARAMETERS:

bInvert; 0=Draw values and labels normally. 1=Invert values and labels.

EXAMPLE:

```
@Y2_INVERT 1
```



PERSISTENT:

NO

@Y2BASE (Y2-Axis Base Line)

For dual-Y and bi-polar charts, this macro specifies a baseline position for the Y2-axis. The default value is 0.0. Values greater than *fBase* draw "Up" from the baseline. Values less than *fBase* draw "Down" from the baseline.

SYNTAX:

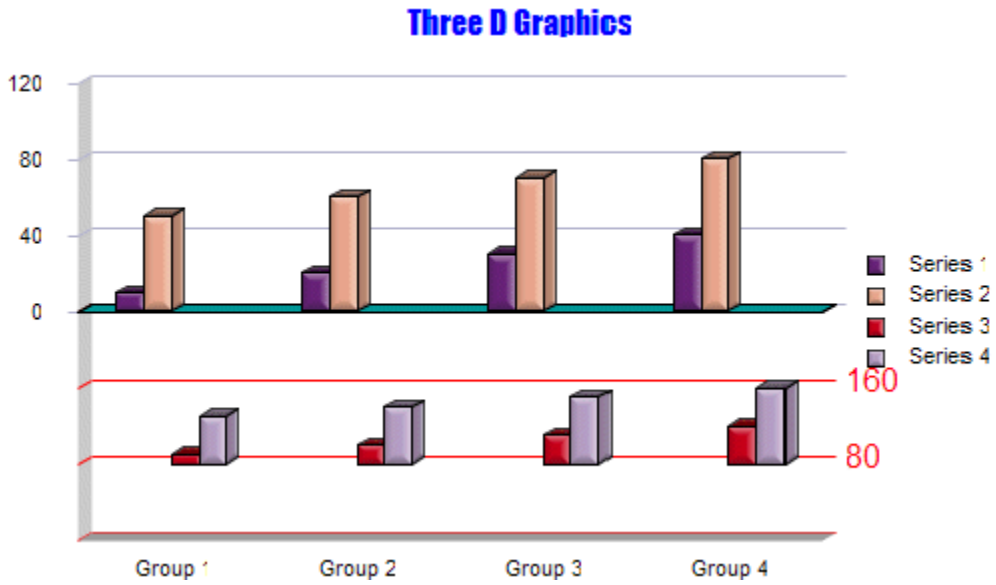
```
@Y2BASE fBase
```

PARAMETERS:

fBase; Base line value

EXAMPLE:

```
@Y2BASE 80
```



PERSISTENT:

NO

Section 4: Series & Groups

These macros can be used to control series and groups.

- @CALC_PERCENT_SERIES; Calculate Percentage Series
- @FORCE_SERIES_COUNT; Force the number of Series in a chart
- @FORECAST; Add Blank Groups to a chart
- @GM; Read data in Column Major or Row Major order
- @IG; Ignore a Group
- @IS; Ignore a Series
- @LIMIT_VISIBLE_GROUPS; Limit the number of Visible Groups
- @RG; Reverse Groups
- @RS; Reverse Series
- @SORT; Sort Series/Groups
- @SWAP; Swap Series/Groups
- @USER_SERIES; Create a User-Defined Series

@CALC_PERCENT_SERIES (Calculate Percent Series)

This macro creates a percentage series by using two other series as the numerator (0...n) and denominator (0...n).

SYNTAX:

```
@CALC_PERCENT_SERIES nNumeratorSeries nDenominatorSeries
szSeriesLabel
```

PARAMETERS:

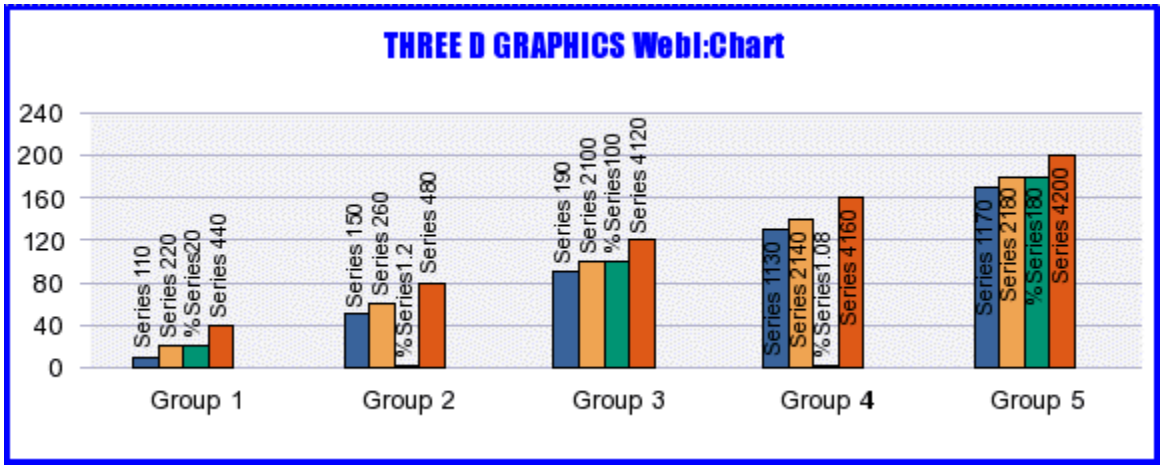
nNumeratorSeries; 0...n selects the numerator series

nDenominatorSeries; 0...n selects the denominator series

szSeriesLabel; new percent series label to use in legend area

EXAMPLE:

```
@CALC_PERCENT_SERIES 1 0 %Series
```



PERSISTENT:

NO

@FORCE_SERIES_COUNT (Force Series Count)

This macro forces the number of series in the chart and number of series drawn in the legend to *nElements*.

SYNTAX:

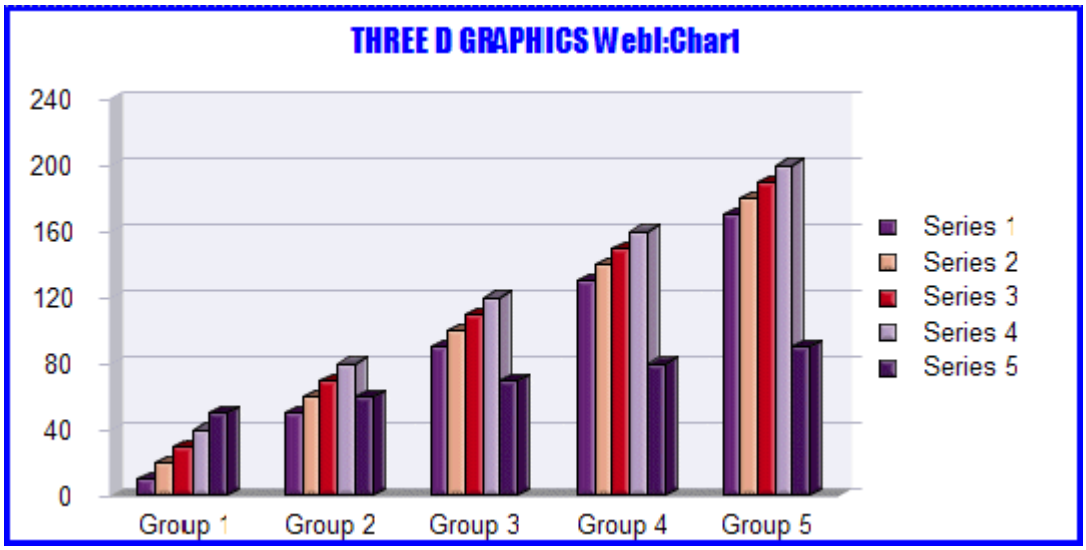
```
@FORCE_SERIES_COUNT nElements
```

PARAMETERS:

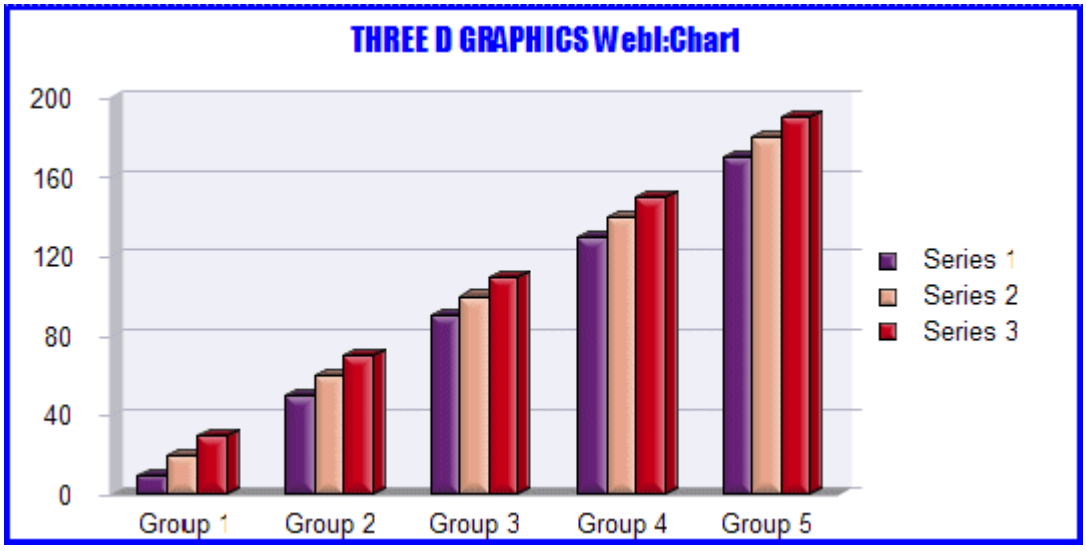
nElements; 0...1024 defines the number of series to draw

EXAMPLE:

```
@FORCE_SERIES_COUNT 5
```




```
@FORCE_SERIES_COUNT 3
```



PERSISTENT

NO

@FORECAST (Add Blank Groups)

This macro adds blank groups to the end of a chart.

SYNTAX:

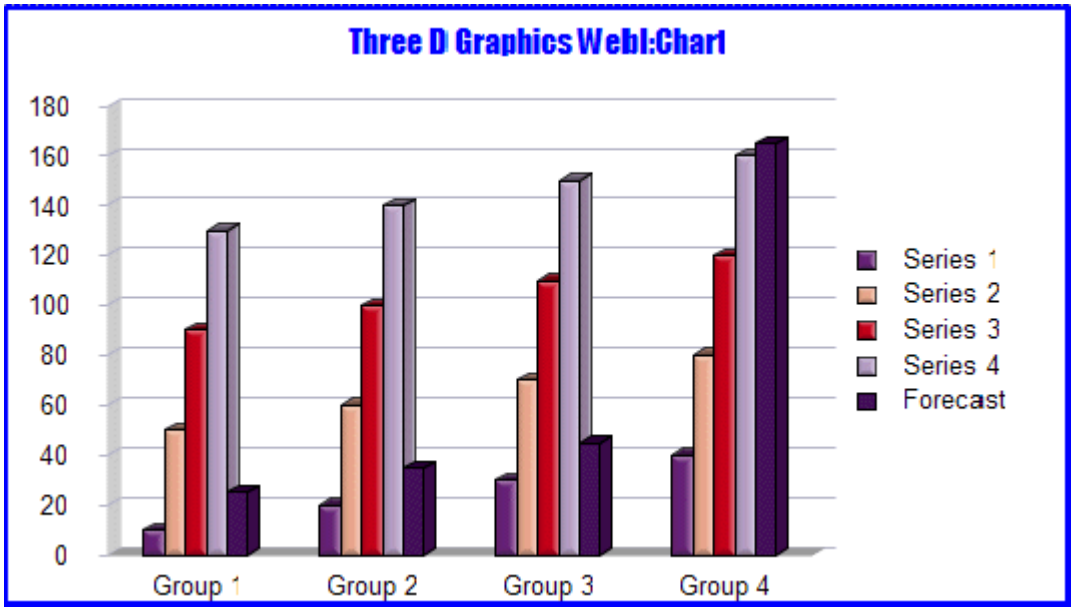
```
@FORECAST nGroups
```

PARAMETERS:

nGroups; 0…1024 groups

EXAMPLE:

```
@FORECAST 1 @DP 0 4 25 @DP 1 4 35 @DP 2 4 45 @DP 3 4 165
@AA @AGL 4 Forecast
```



PERSISTENT:

NO

ALSO SEE:

@DP (to assign data points to the new blank groups)

@GM (Group Major)

This macro reads from the internal data matrix in Column Major order instead of the default Row Major order.

SYNTAX:

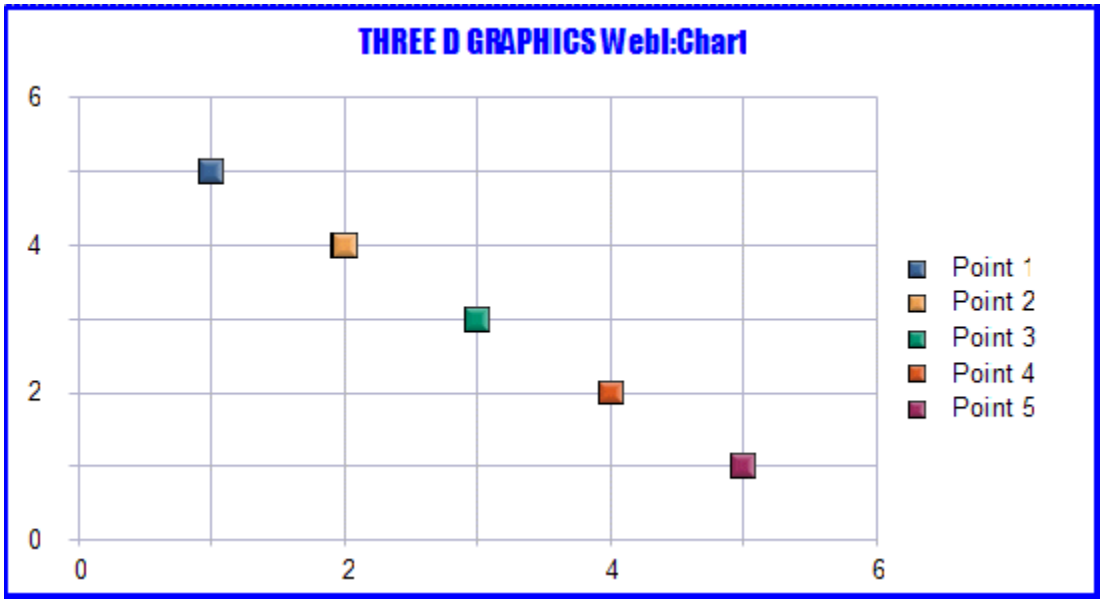
```
@GM bVertFormat
```

PARAMETERS:

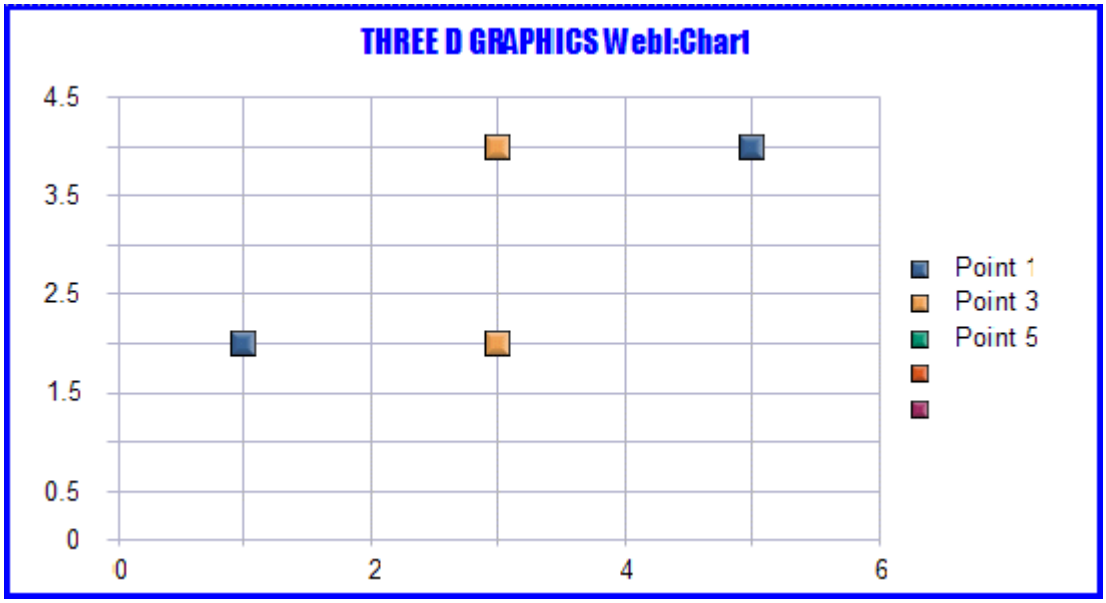
bVertFormat; 1 = true (read data in column major order), 0 = false (read data in row major order)

EXAMPLE:

```
@GM 0
```




```
@GM 1
```



PERSISTENT:

NO

@IG (Ignore Group)

This macro hides a specified group and its risers so that it is not drawn in the chart.

SYNTAX:

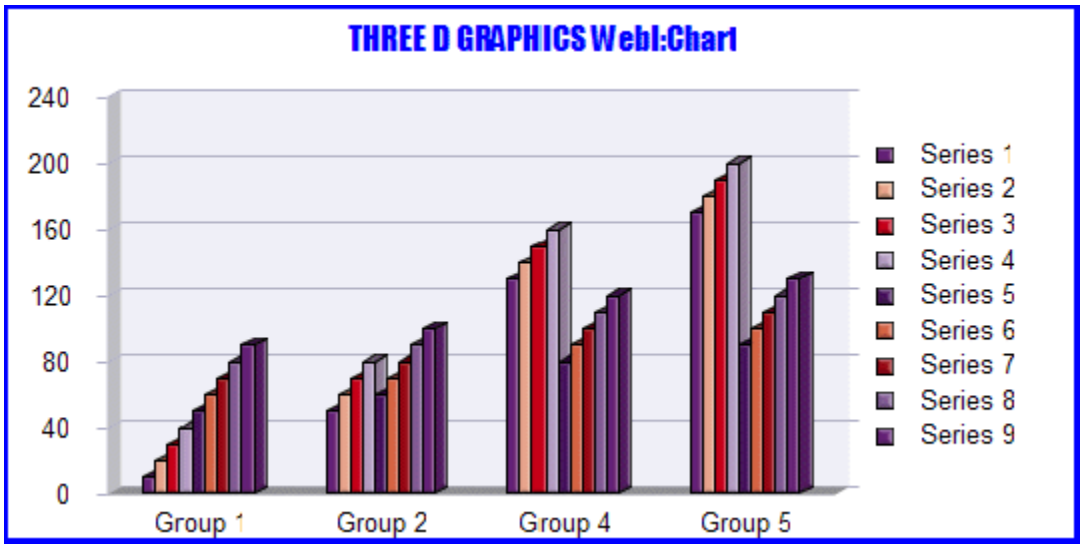
```
@IG nGroup
```

PARAMETERS:

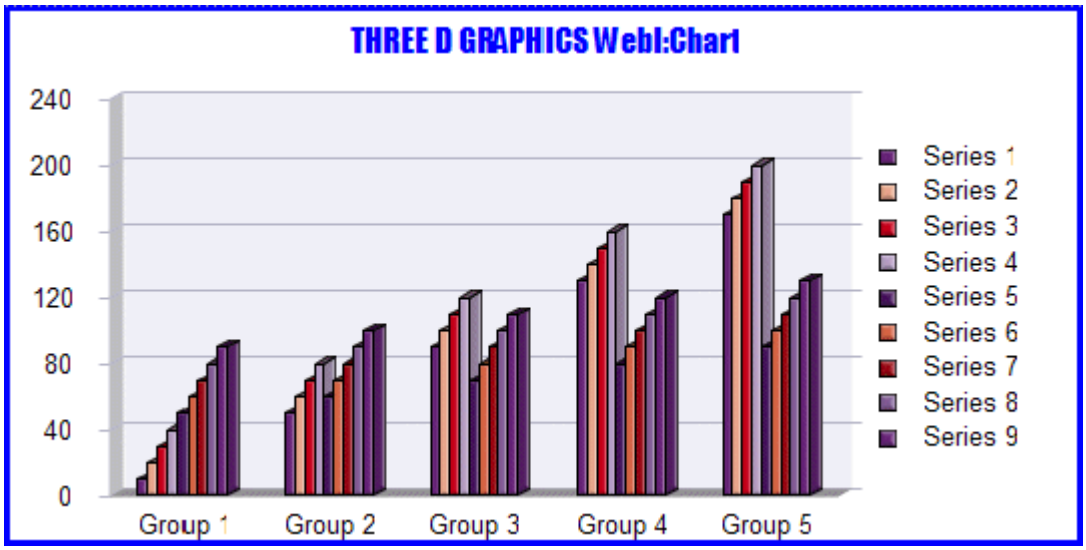
nGroup; -1...1024 specifies the group to hide. -1=Restore all previously ignored groups, 0 = Hide Group 1, 1 = Hide Group 2, etc.

EXAMPLE:

```
@IG 2
```




```
@IG -1
```



PERSISTENT:

YES

@IS (Ignore Series)

This macro sets the specified series *nSeries* to "ignore" so that it will not appear in the chart. The special value of -1 "restores" all series so that they will all appear again.

SYNTAX:

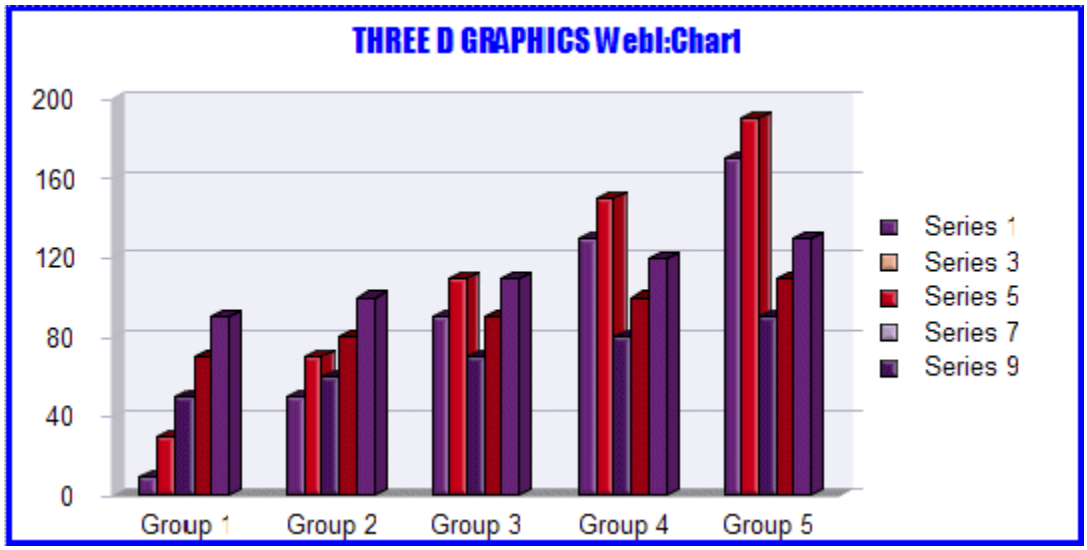
```
@IS nSeries
```

PARAMETERS:

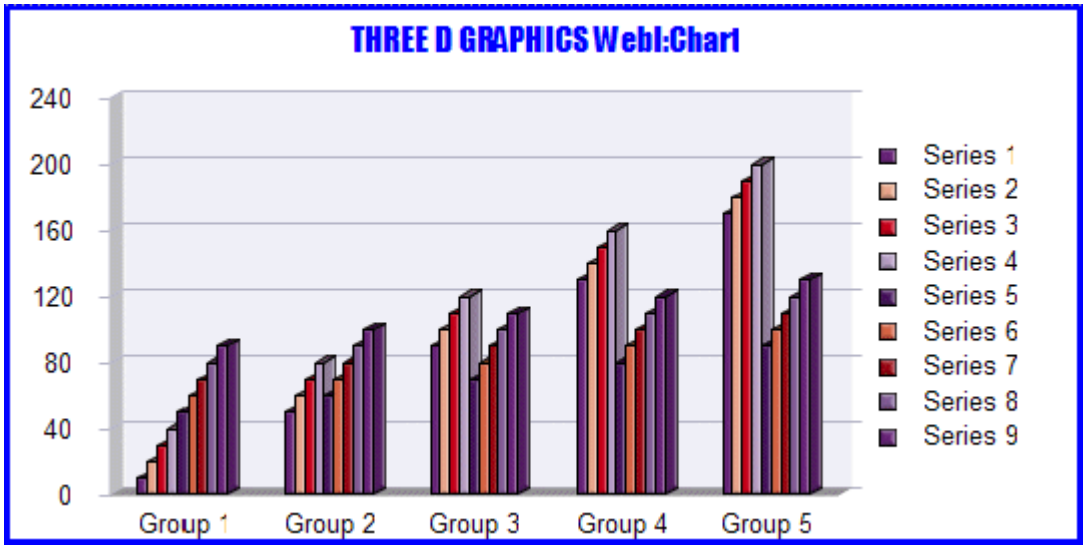
nSeries; -1...1024 specifies the series to hide. -1=Restore all previously ignored Series, 0 = Hide Series 1, 1 = Hide Series 2, etc.

EXAMPLE:

```
@IS 1
```




```
@IS -1
```



PERSISTENT:

YES

@LIMIT_VISIBLE_GROUPS (Limit Visible Groups)

This macro limits the number of visible groups in a chart by suppressing all data after *nGroup*. This macro is very useful for area charts where you wish to "cut off" data at some point in the chart.

SYNTAX:

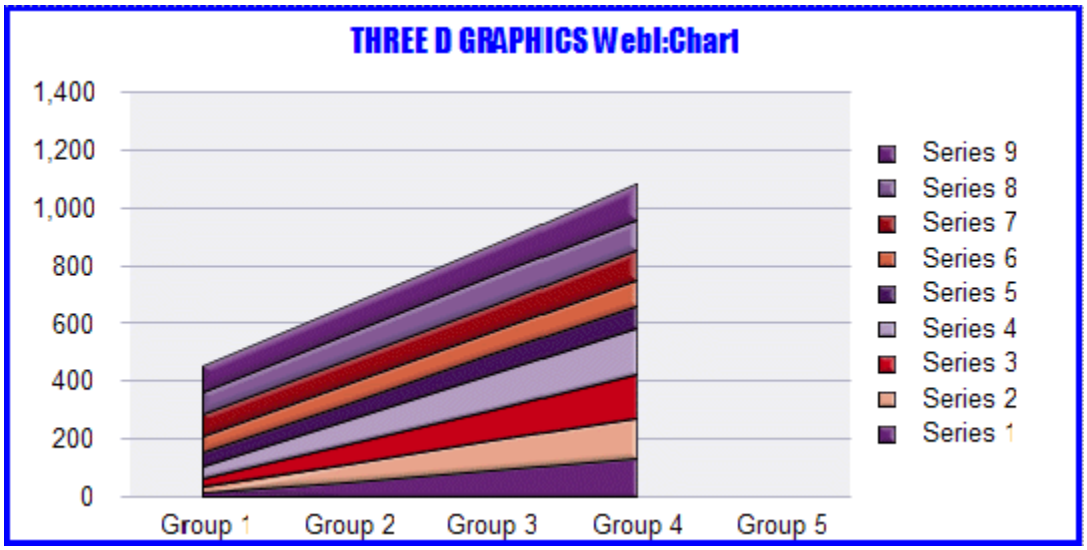
```
@LIMIT_VISIBLE_GROUPS nGroup
```

PARAMETERS:

nGroup; 0..1024. 0 = "no suppression" 1...1024 = number of groups that will be visible

EXAMPLE:

```
@LIMIT_VISIBLE_GROUPS 4
```



PERSISTENT:

YES

NOTES:

This macro only works for Single-Y stacked chart types:

@GRAPHTYPE Value	Chart
1	Vertical Area Stacked
8	Horizontal Area Stacked
15	Vertical Bar Stacked
22	Horizontal Bar Stacked
29	Vertical Line Stacked
36	Horizontal Line Stacked

@RG (Reverse Group)

This macro reverses the order of groups in a chart.

SYNTAX:

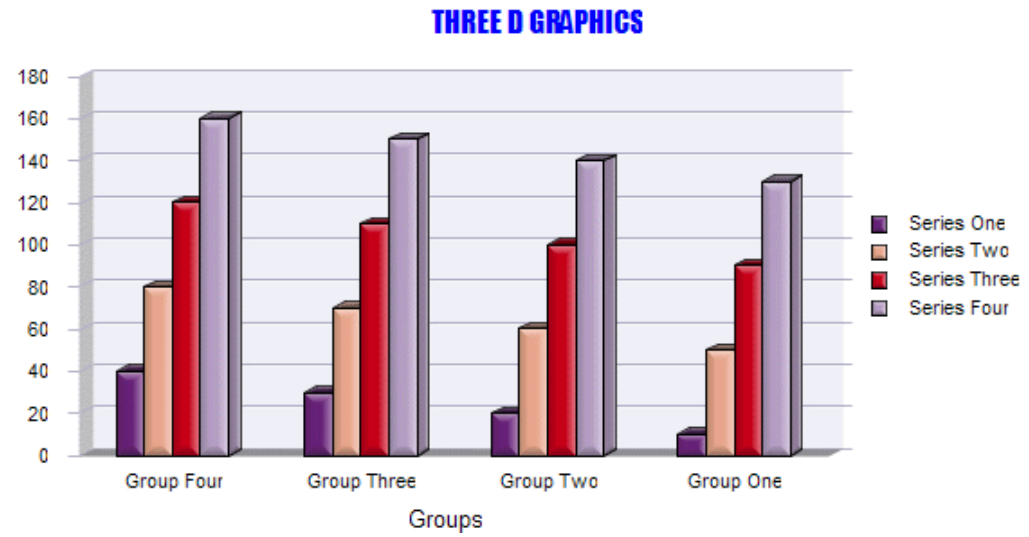
```
@RG bColReverse
```

PARAMETERS:

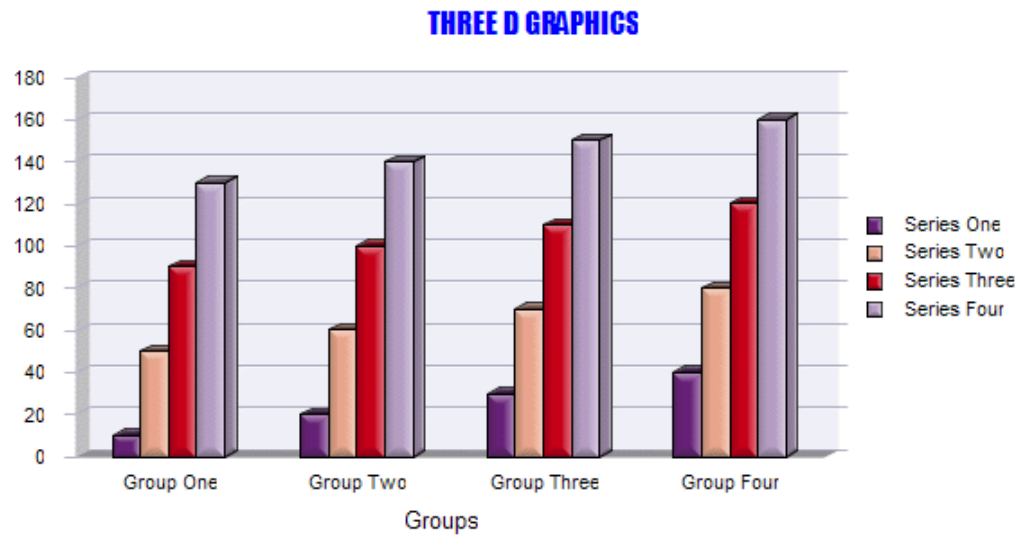
bColReverse; 1=true/reverse groups, 0=false/do not reverse groups.

EXAMPLE:

```
@RG 1
```




```
@RG 0
```



PERSISTENT:

YES

ALSO SEE:

@RS

@RS (Reverse Series)

This macro reverses the order of series in a chart.

SYNTAX:

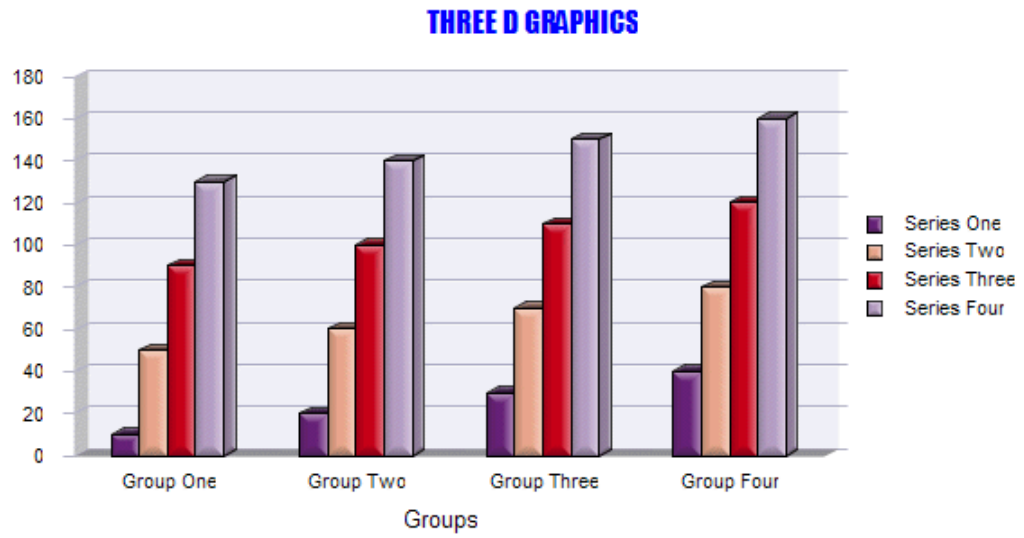
```
@RS bRowReverse
```

PARAMETERS:

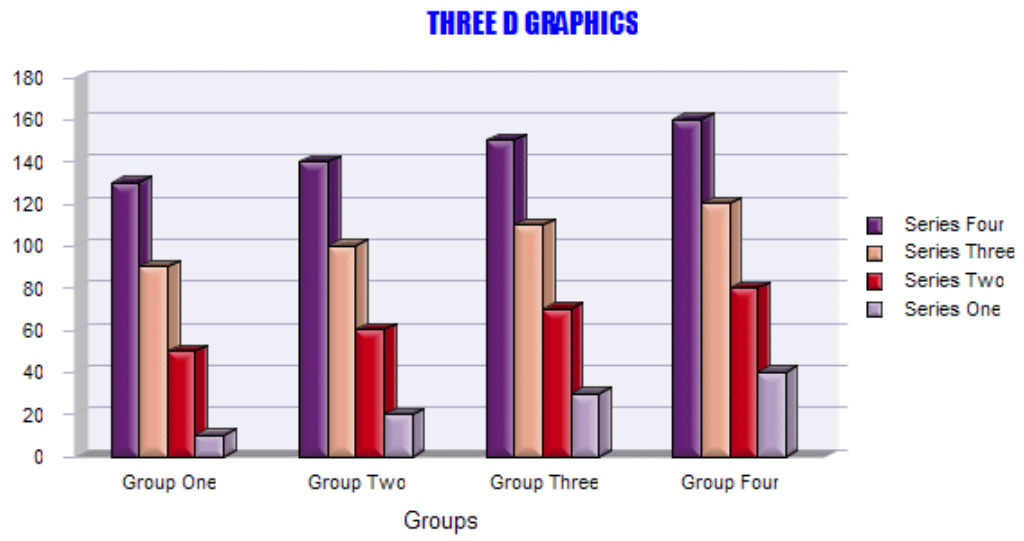
bRowReverse; 1=true/reverse series, 0=false/do not reverse series.

EXAMPLE:

```
@RS 0
```




```
@RS 1
```



PERSISTENT:

YES

ALSO SEE:

@RG

@SORT (Sort Series/Groups)

This macro can be used to sort series or groups in a chart.

SYNTAX:

```
@SORT nSort
```

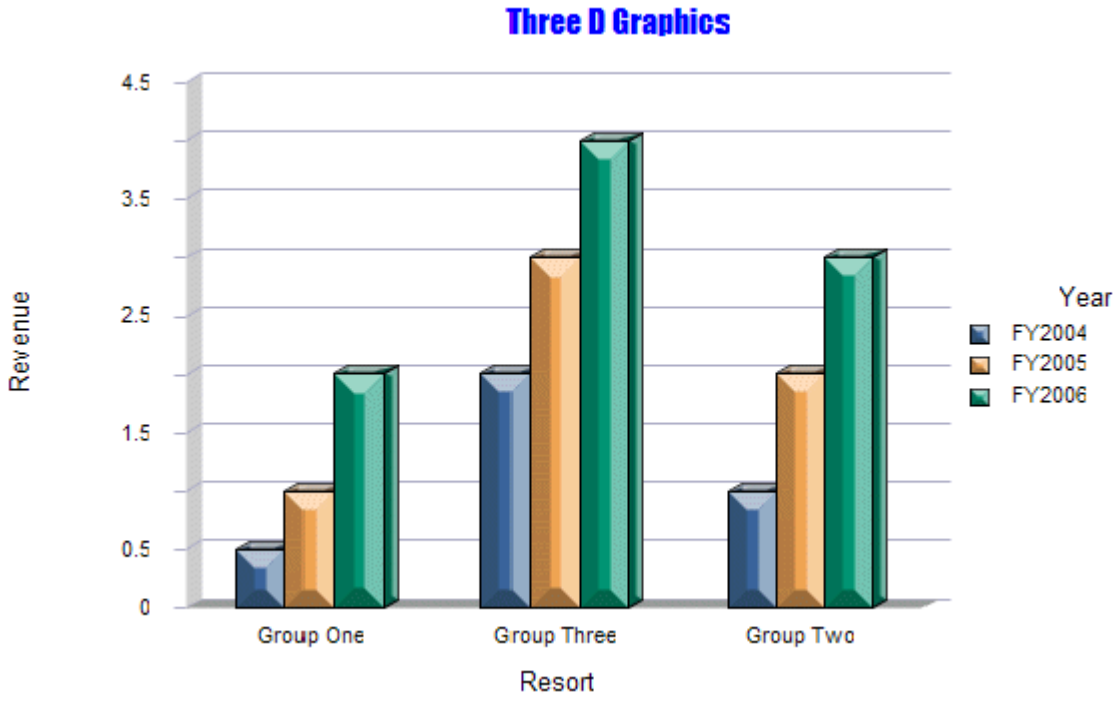
PARAMETERS:

nSort; 0...9 selects one of the following sorting options:

nSort	Sorting Option
0	Sort Series Labels in Alphabetical order (a...z)
1	Sort Series Labels in Reverse Alphabetical order (z...a)
2	Sort Groups Labels in Alphabetical order (a...z)
3	Sort Groups Labels in Reverse Alphabetical order (z...a)
4	Sort Series Totals in ascending numeric value (i.e. total all values in each series. Then 'rank' the series from smallest total to largest).
5	Sort Series Totals in descending numeric value
6	Sort Groups Totals in ascending numeric value
7	Sort Groups Totals in descending numeric value
8	First Series Key to Groups Ascending
9	First Series Key to Groups Descending

EXAMPLE:

```
@SORT 2
```



PERSISTENT:

NO

@SWAP (Swap Series/Groups)

This macro can be used to swap series and group orientation.

SYNTAX:

```
@SWAP bSwap
```

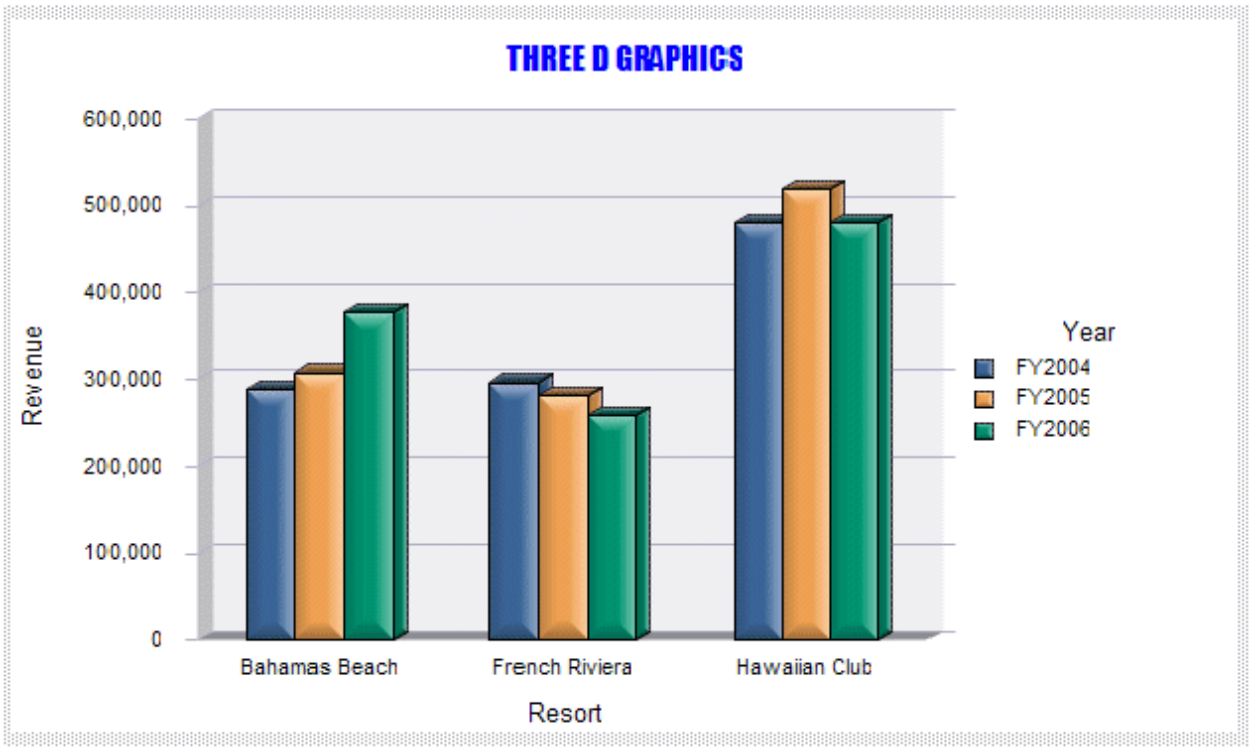
PARAMETERS:

bSwap; 1 = swap series/group orientation, 0= do not swap series/group orientation

EXAMPLE:

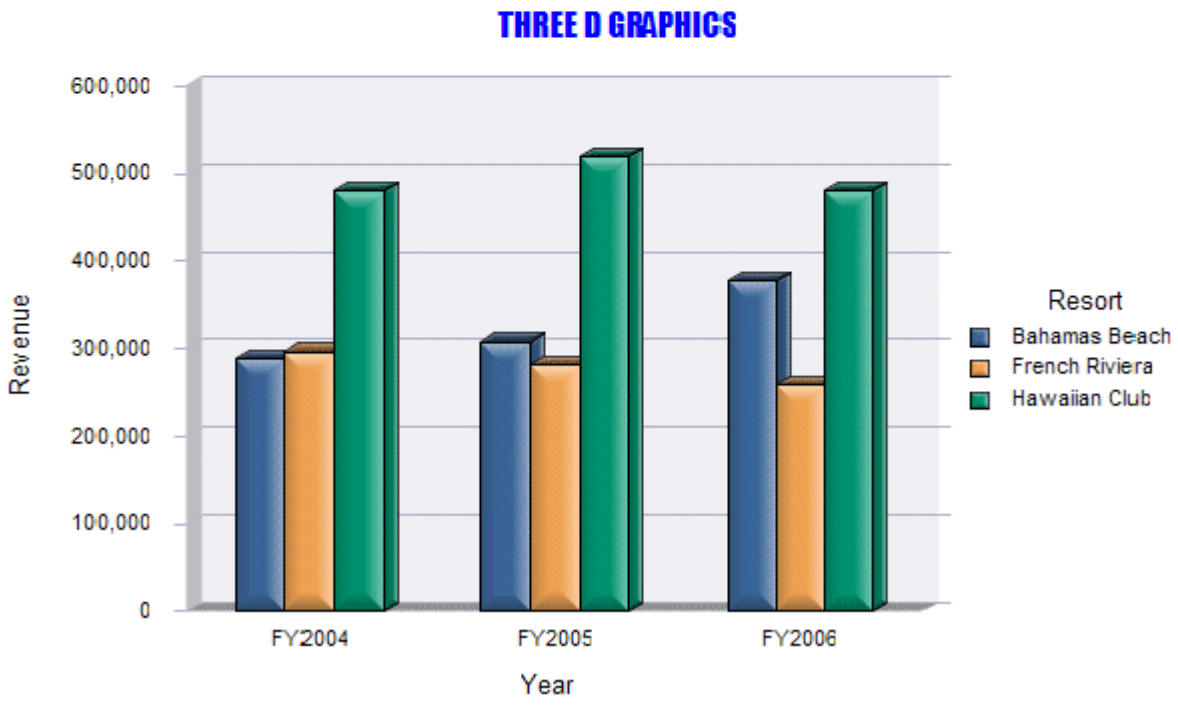
```
@SWAP 0
```

Year	Resort	Revenue
FY2004	Bahamas Beach	287,929
FY2004	French Riviera	295,940
FY2004	Hawaiian Club	479,685
FY2005	Bahamas Beach	307,400
FY2005	French Riviera	280,310
FY2005	Hawaiian Club	519,530
FY2006	Bahamas Beach	376,115
FY2006	French Riviera	259,170
FY2006	Hawaiian Club	480,445




```
@SWAP 1
```

Year	Resort	Revenue
FY2004	Bahamas Beach	287,929
FY2004	French Riviera	295,940
FY2004	Hawaiian Club	479,685
FY2005	Bahamas Beach	307,400
FY2005	French Riviera	280,310
FY2005	Hawaiian Club	519,530
FY2006	Bahamas Beach	376,115
FY2006	French Riviera	259,170
FY2006	Hawaiian Club	480,445



PERSISTENT:

YES

@USER_SERIES (User-Defined Series)

This macro can be used to define an arbitrary series of your own making. It will be appended to the end of the data coming from the data set and will, therefore, always be the last series in the legend.

SYNTAX:

```
@USER_SERIES nElements fValue1 fValue2 ... fValueN szSeriesName
```

PARAMETERS:

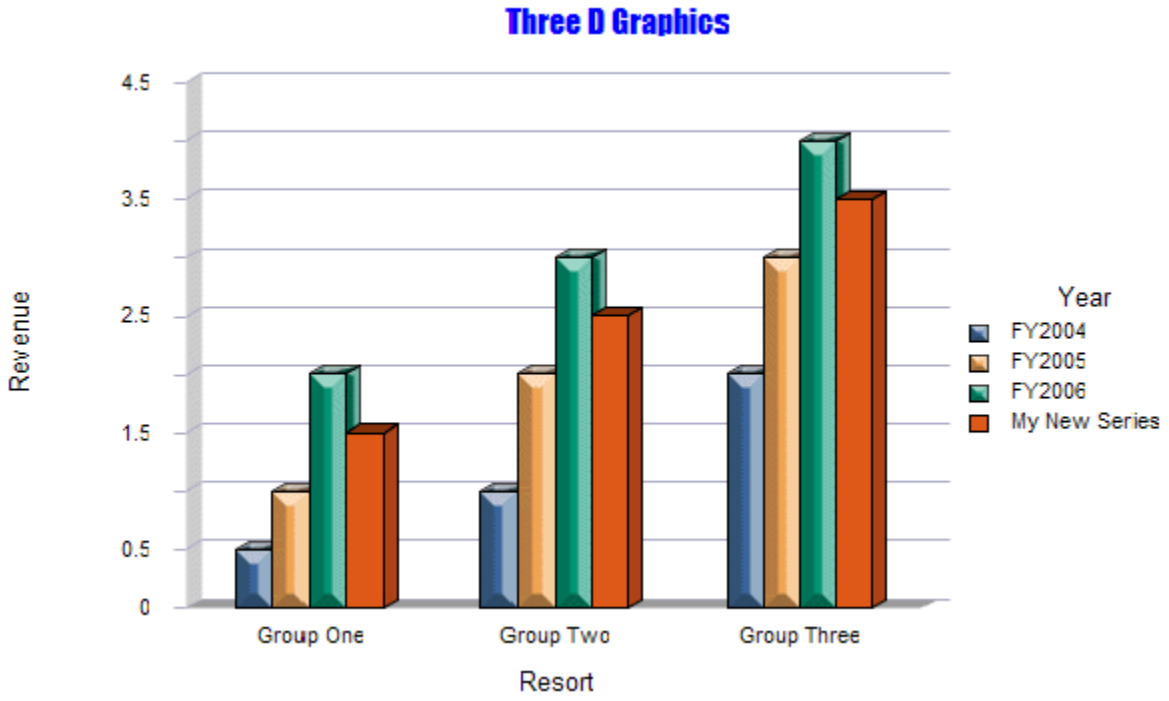
nElements; 1...1024 defines the number of *fValues* that follow. For example if *nElements* is 3, it must be followed by 3 *fValues* that will be assigned to the first 3 groups of the new series.

fValue1 fValue2 ... fValueN; Values to assign to each *nElements.*

szSeriesName; Name of the new series what will appear in the legend. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_SERIES 3 1.5 2.5 3.5 My New Series
```



PERSISTENT:

NO

Section 5: Labels

Use these macros to manage series and group labels.

- @3DLABEL; Adjust labels in a 3D Chart.
- @AGL; Alias/Change a Group Label
- @ASL; Alias/Change a Series Label
- @COND_GROUP_LABEL; Color a riser based on a Group Label
- @COND_GROUP_LABEL2; Color a riser based on a Group Label prefix
- @XSKIP; Skip labels on the X-Axis
- @XSKIP2; Skip labels on the X-Axis and force last label

@3DLABEL (Adjust Labels in a 3D Chart)

This macro can be used to change the position of labels in a 3D chart.

SYNTAX:

```
@3DLABEL nLabel nX nY
```

PARAMETERS:

nLabel = 0...3 selects which labels to adjust.

0=group labels (normally along the right floor of the 3D cube)

1=series labels (normally along the left floor of the 3D cube)

2=labels on the left side of the 3D cube

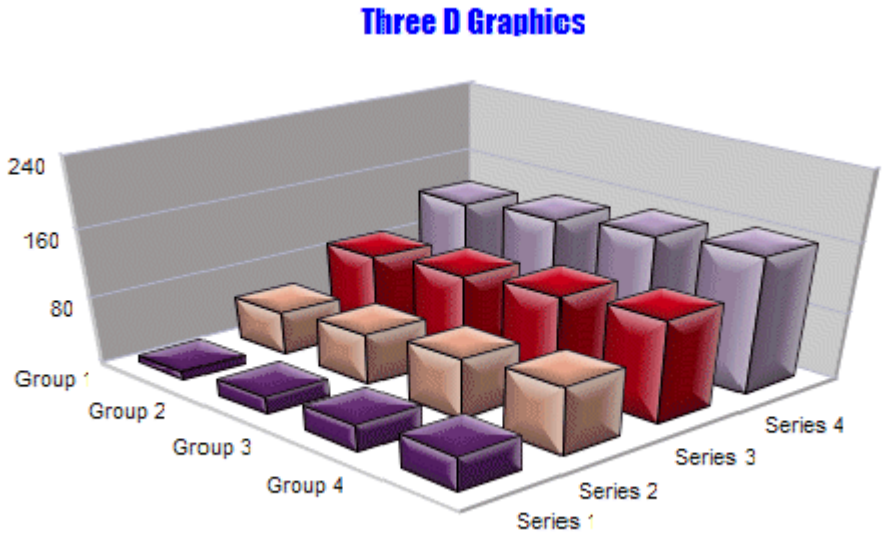
3=labels on the right side of the 3D cube.

nX = -2000....2000 selects the amount of adjustment in the X-direction. If *nLabel* is 2 or 3, negative values move labels up and positive values move labels down. If *nLabel* is 0 or 1, negative values move labels closer to the cube and positive values move labels further away from the cube.

nY = -2000....2000 selects the amount of adjustment in the Y-direction. Negative values move labels closer to the cube. Positive values move labels further away from the cube.

EXAMPLE:

```
@3DLABEL 0 1 1
```



PERSISTENT:

YES

@AGL (Alias Group Label)

This macro can be used to change a group's label.

SYNTAX:

```
@AGL nGroup szLabel
```

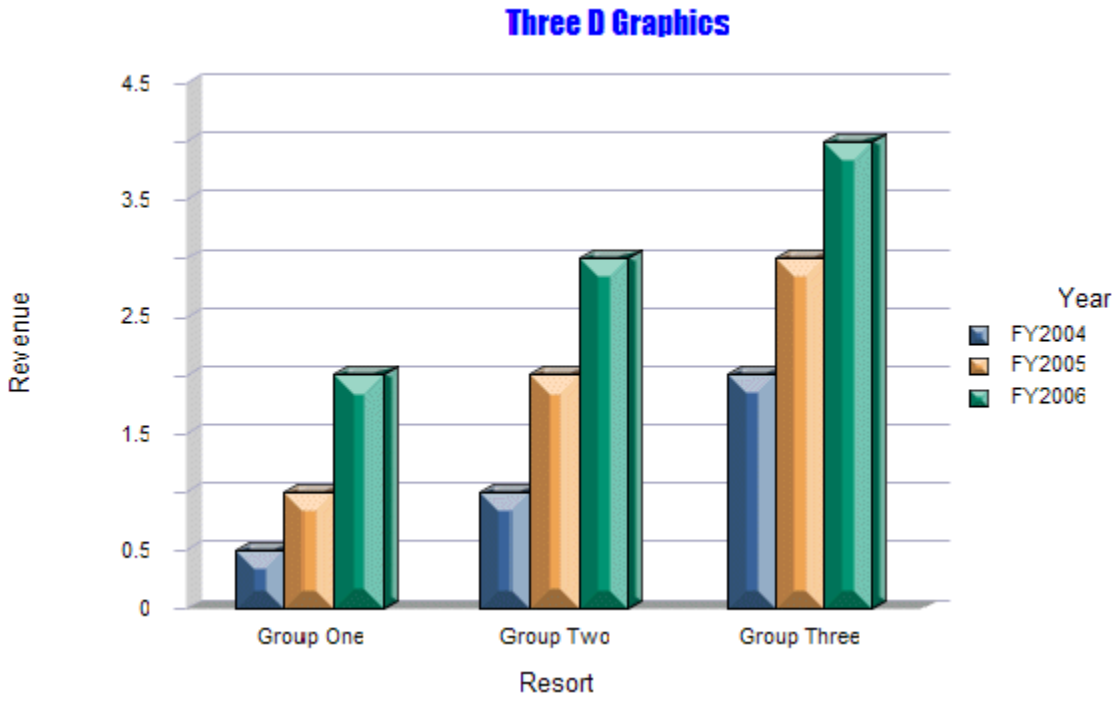
PARAMETERS:

nGroup; 0...number of groups in chart (0=Group 1). Defines the group on which to place the alias label.

szLabel; Group label. If you intend to define another macro in the same title field, terminate the label string with a '~'.

EXAMPLE:

```
@AGL 0 Group One~ @AGL 1 Group Two~ @AGL 2 Group Three
```



PERSISTENT:

NO

ALSO SEE:

@ASL

@ASL (Alias Series Label)

This macro can be used to change a series label.

SYNTAX:

```
@ASL nSeries szLabel
```

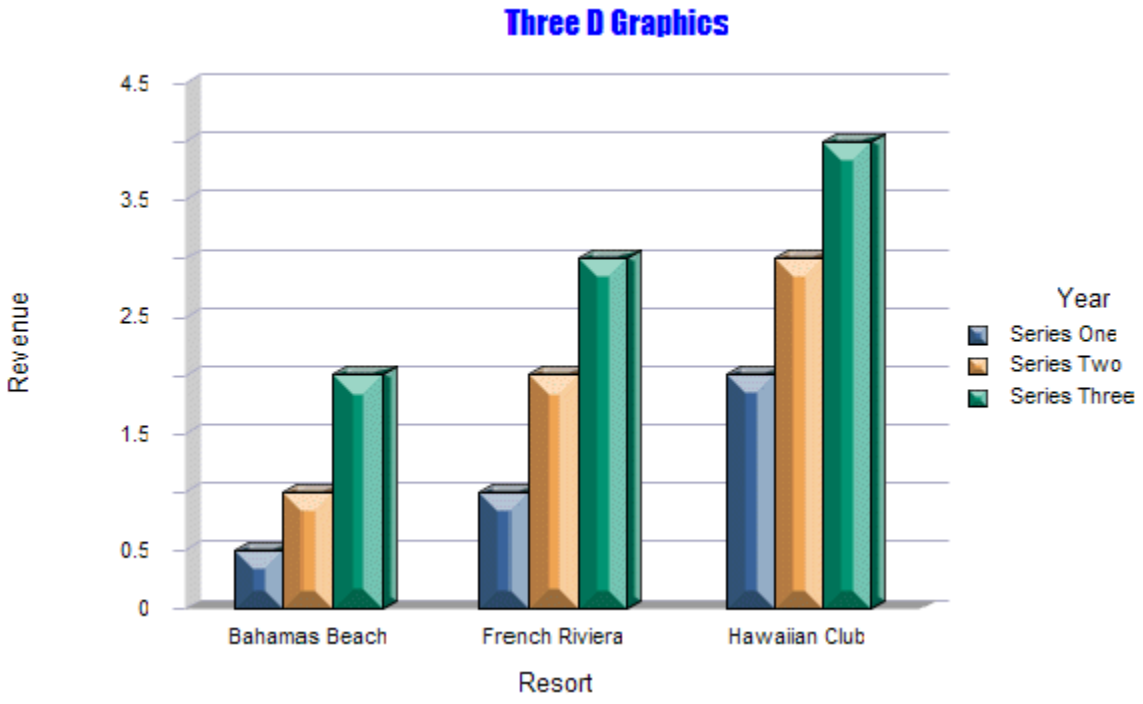
PARAMETERS:

nSeries; 0...number of series in chart (0=Series 1). Defines the series on which to place the alias label.

szLabel; Series label. If you intend to define another macro in the same title field, terminate the label string with a '~'.

EXAMPLE:

```
@ASL 0 Series One~ @ASL 1 Series Two~ @ASL 2 Series Three
```



PERSISTENT:

NO

ALSO SEE:

@AGL

@COND_GROUP_LABEL (Conditional Group Label)

This macro will apply a color to the riser(s) at *nSeries* if the series' group label is *szGroupLabel*.

SYNTAX:

```
@COND_GROUP_LABEL nSeries nRed nGreen nBlue szGroupLabel
```

PARAMETERS:

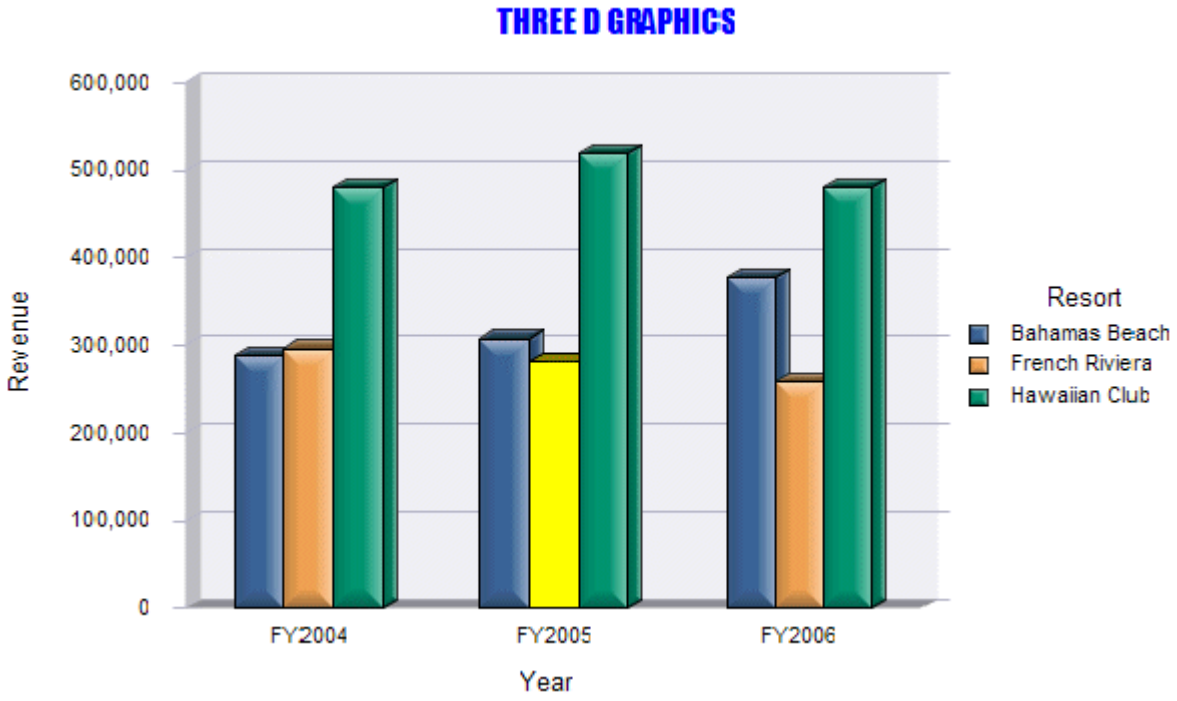
nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nRed, nGreen, nBlue; 0...255 color to use for series riser.

szGroupLabel; Group label string

EXAMPLE:

```
@COND_GROUP_LABEL 1 255 255 0 FY2005
```



PERSISTENT:

NO

NOTES:

- The color will only be applied to the first instance of *szGroupLabel* in the group labels. If two groups have the same label, the color is only applied to the first instance. The *szGroupLabel* cannot be specified as a runtime parameter (i.e., P3). It must be a literal string.
- The group label string (*szGroupLabel*) is case sensitive ("GROUP 1" will not match "Group 1").

@COND_GROUP_LABEL2 (Conditional Group Label 2)

This macro will apply a color to the riser(s) at *nSeries* when a group label prefix matches the group label. To use this macro, you must prefix the target label and a tilde (~) to each group label. When the macro finds a group label that matches the prefix, the color is applied to the riser.

SYNTAX:

```
@COND_GROUP_LABEL2 nSeries nRed nGreen nBlue
```

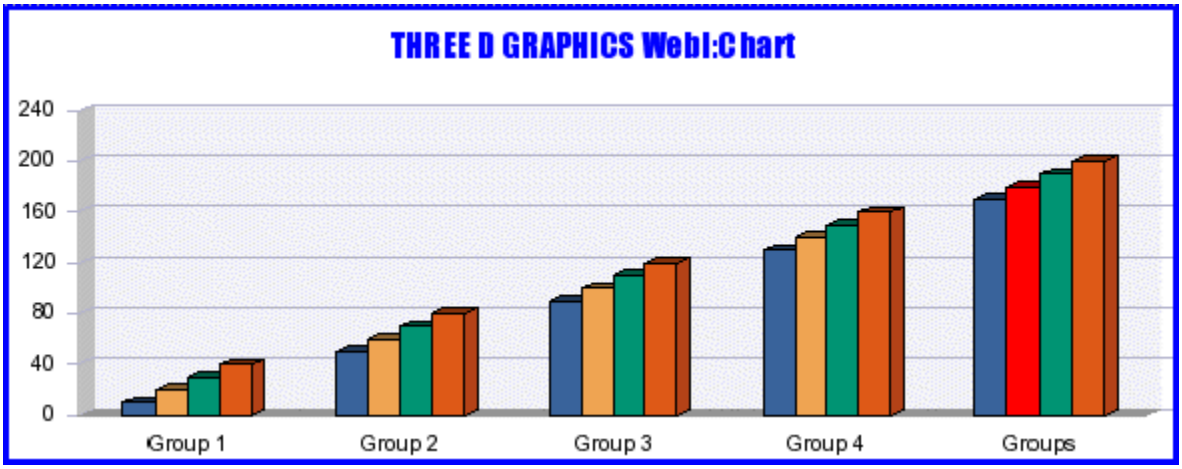
PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nRed, nGreen, nBlue; 0...255 color to use for series riser.

EXAMPLE:

```
@COND_GROUP_LABEL2 1 255 0 0
```



In this example, each group label has a “Groups~” prefix.

PERSISTENT:

NO

@XSKIP (X-Axis Skip Labels)

This macro specifies an interval at which to skip labels on a group or X-axis.

SYNTAX:

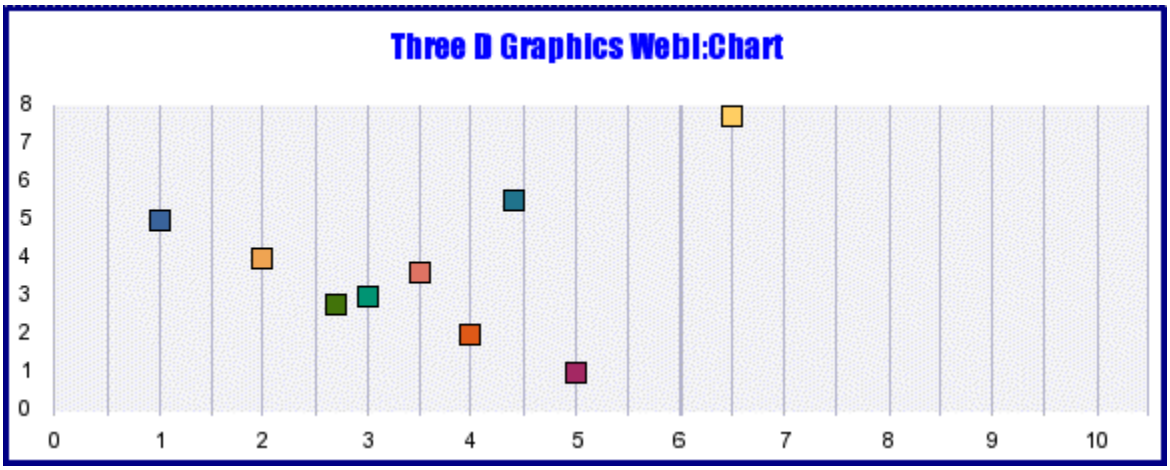
```
@XSKIP nSkip
```

PARAMETERS:

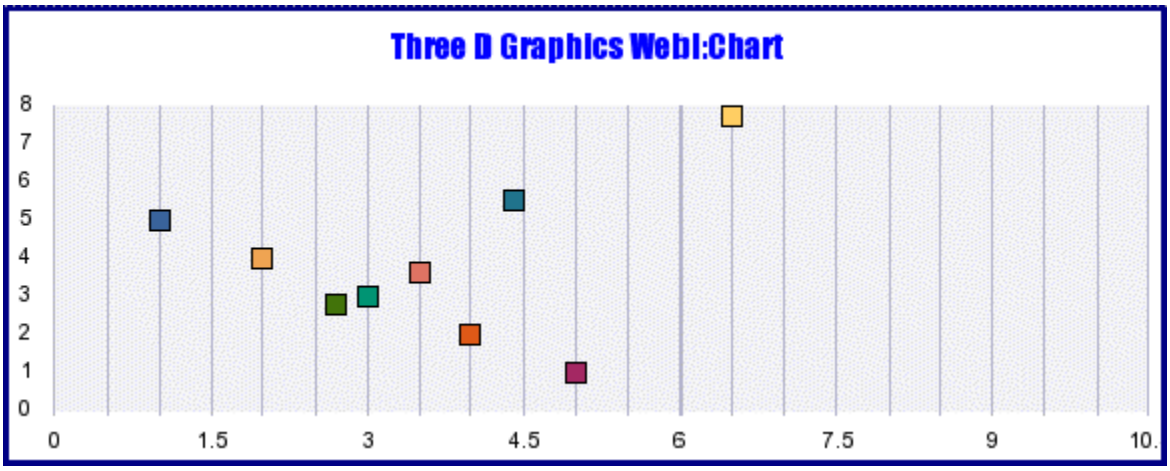
nSkip; skip value

EXAMPLE:

```
@XSKIP 1
```




```
@XSKIP 2
```



PERSISTENT:

NO

ALSO SEE:

@XSKIP2

@XSKIP2 (X-Axis Skip/Force Last Label)

This macro is the same as the @XSKIP macro except it forces the last label to be visible regardless of the skip value.

SYNTAX:

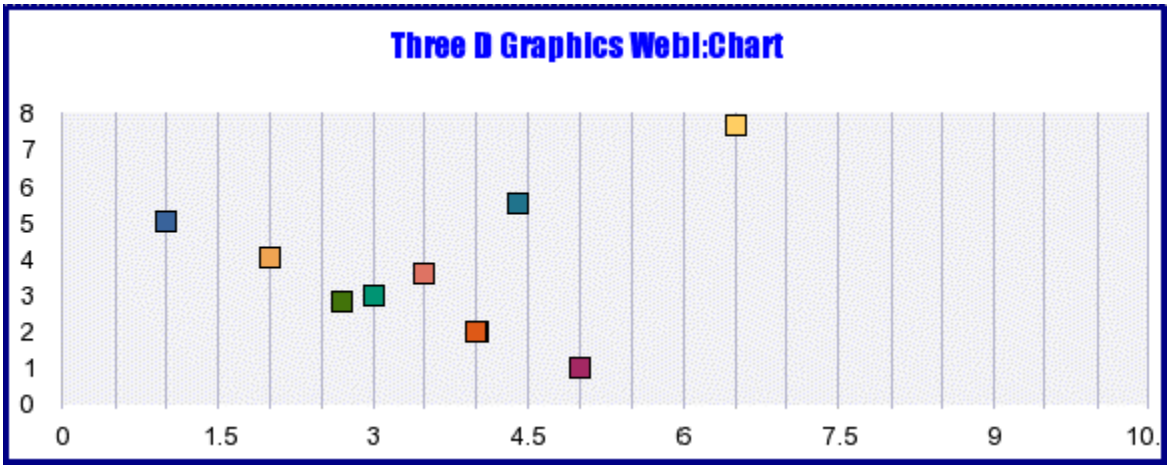
```
@XSKIP2 nSkip
```

PARAMETERS:

nSkip; skip value

EXAMPLE:

```
@XSKIP2 2
```



PERSISTENT:

NO

ALSO SEE:

@XSKIP, @XSKIP3

Section 6: Riser & Markers

Use these macros to control and format risers and markers:

- @GAP; Gap between groups of risers
- @MARKER; Define Marker Shapes
- @MCOLOR; Define Marker Colors
- @PAT; Apply a Pattern to Riser
- @RISER_BORDER; Enable/Disable Riser Borders
- @RISER_OVERLAP; Define the amount of overlap between risers in side-by-side/clustered bar charts
- @RISER_WIDTH; Define the Width of Risers in bar charts
- @SZ; Marker Size

@GAP (Gap between Riser Groups)

This macro defines a gap between riser groups.

SYNTAX:

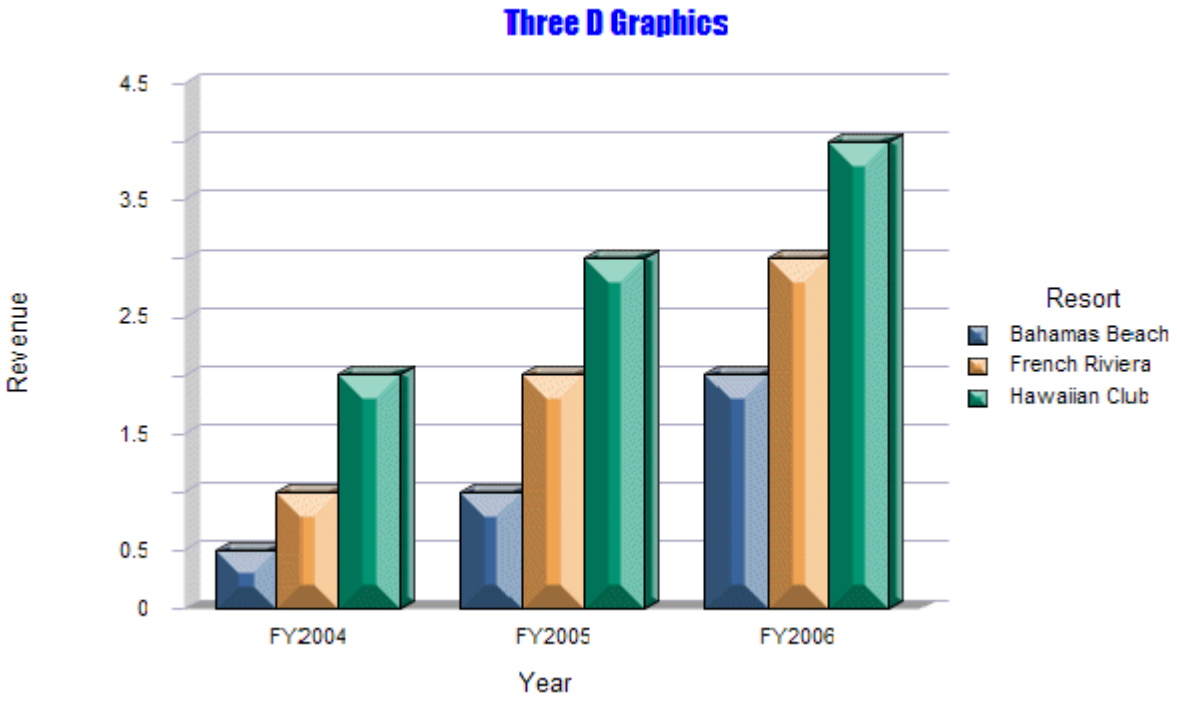
```
@GAP nGAP
```

PARAMETERS:

nGap; 0...100 gap between riser groups

EXAMPLE:

```
@GAP 25
```



@MARKER (Marker Shapes)

This macro sets the shape of markers for a particular series in a chart. It can be used in any chart that uses markers (Bubble, Scatter, Line Graph with Markers, etc.) except Box Plots. See the @MS macro to set the shape of markers in box plots.

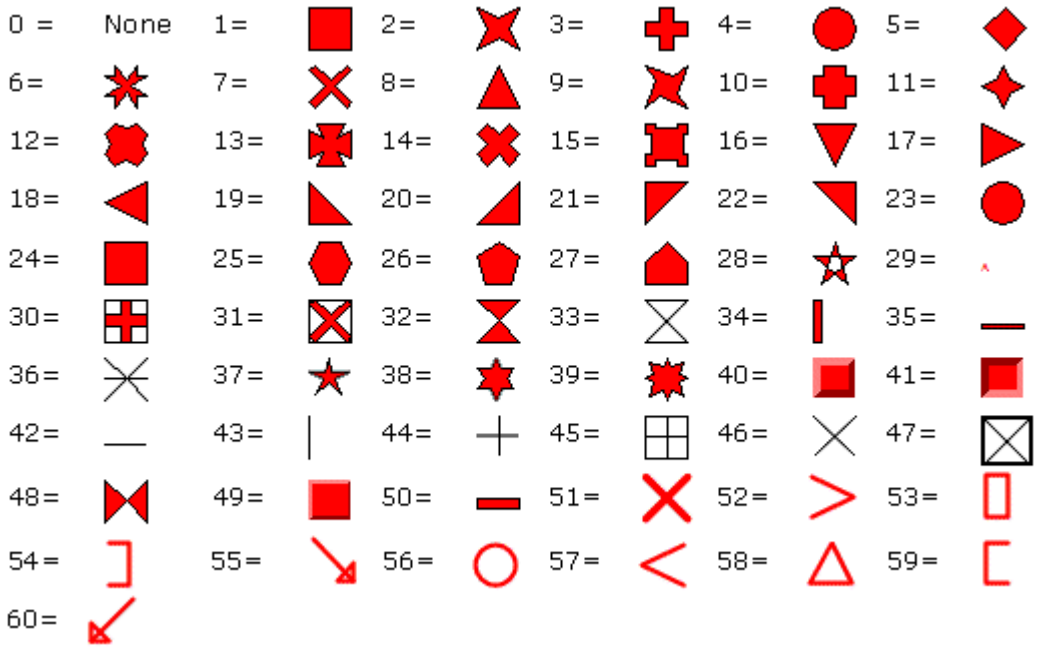
SYNTAX:

```
@MARKER nSeries nMarker
```

PARAMETERS:

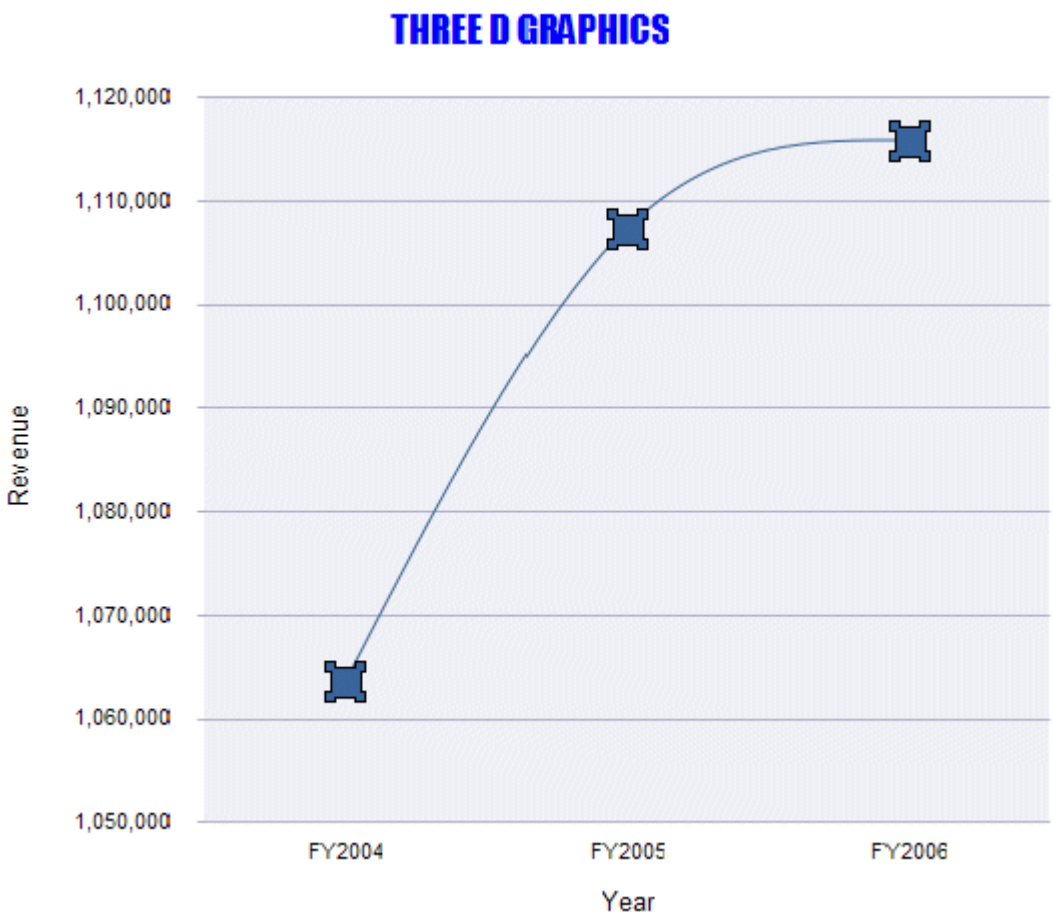
nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nMarker; 0...60 selects one of the following markers to apply to *nSeries*.



EXAMPLE:

```
@MARKER -1 15
```



PERSISTENT:

YES

@MCOLOR (Marker Colors)

This macro can be used to change the color of markers and risers in all chart types except box plots. Use the @MC macro if you want to change the color of markers in box plots.

SYNTAX:

```
@MCOLOR nSeries nRed nGreen nBlue
```

PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1=apply to all series, 0=Series 1, 1=Series 2, etc.

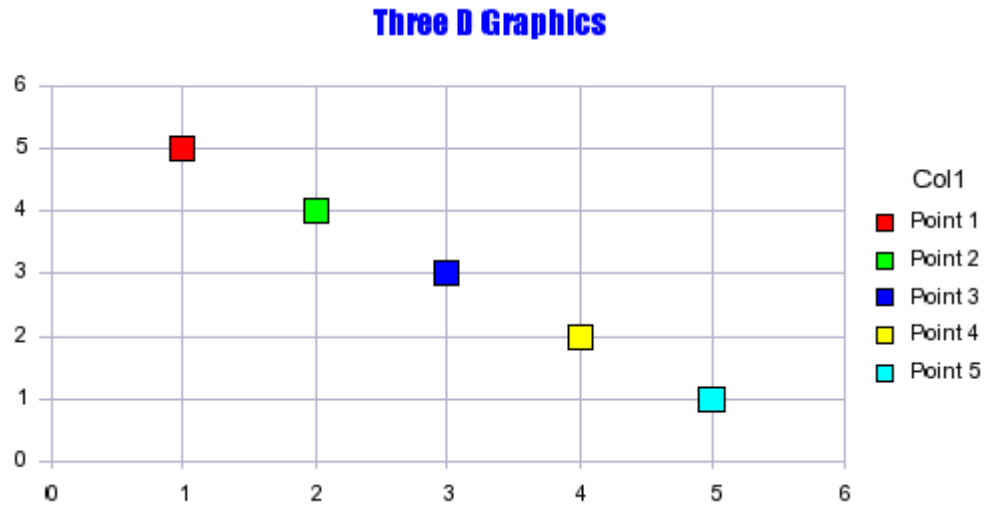
nRed; 0...255 defines the Red portion of RGB color selection.

nGreen; 0...255 defines the Green portion of RGB color selection.

nBlue; 0...255 defines the Blue portion of RGB color selection.

EXAMPLE:

```
@MCOLOR 0 255 0 0 @MCOLOR 1 0 255 0 @MCOLOR 2 0 0 255
```



PERSISTENT:

YES

ALSO SEE:

@GCOLOR to change the color of other chart objects.

@PAT (Riser Pattern)

This macro can be used to apply a pattern to risers.

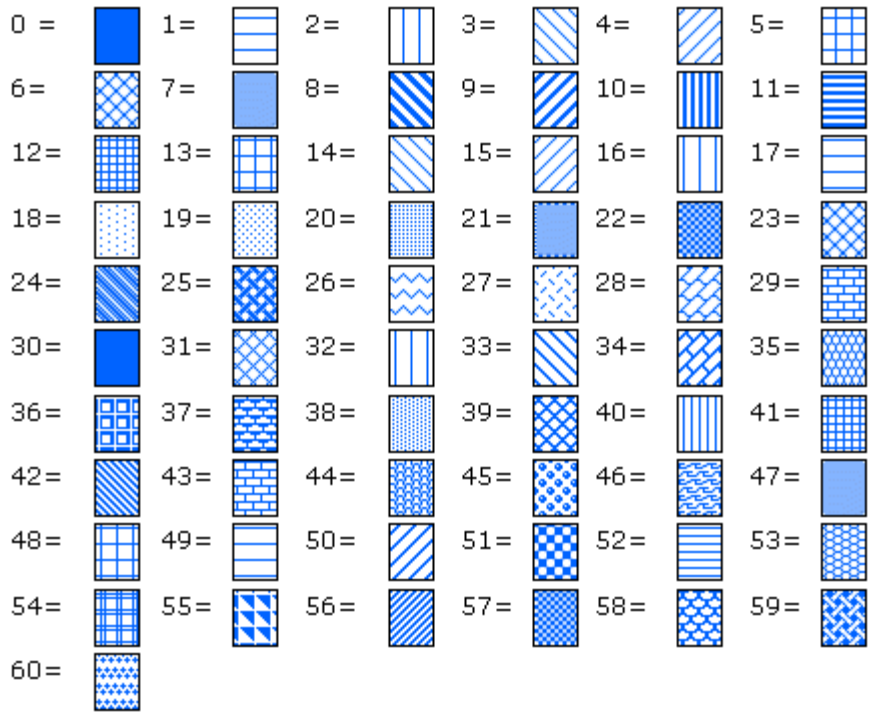
SYNTAX:

```
@PAT nSeries nPattern
```

PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

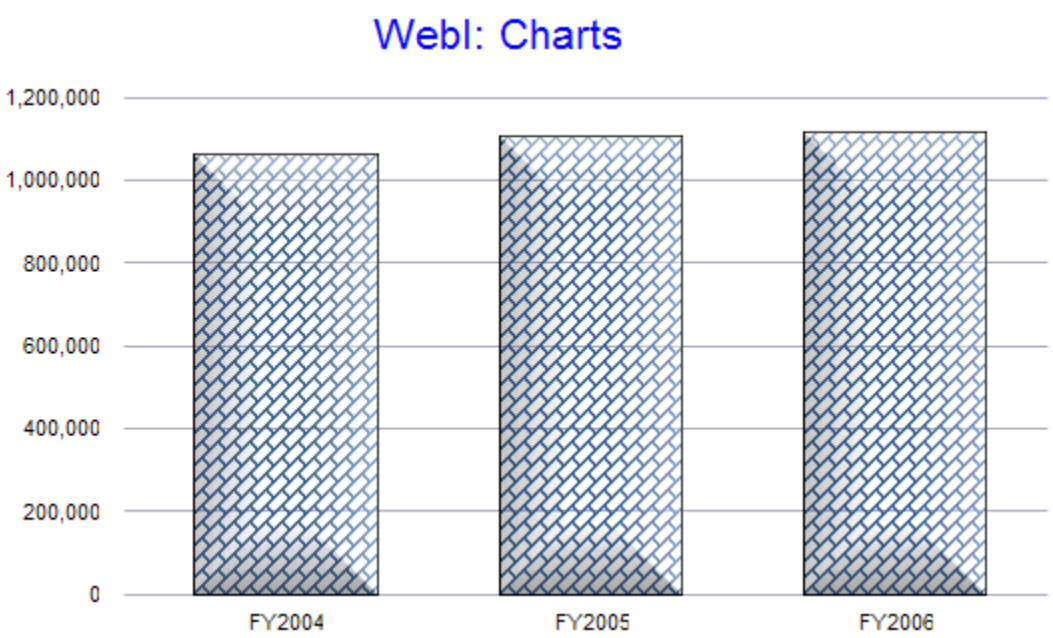
nPattern; -60...60. Use 0...60 to select one of the following patterns with a white background.



Use –1 to –60 to select one of these patterns with a transparent background.

EXAMPLE:

```
@PAT -1 -28
```



PERSISTENT:

YES

@RISER_BORDER (Riser Border)

This macro enables/disables drawing of borders around risers/markers in a chart.

SYNTAX:

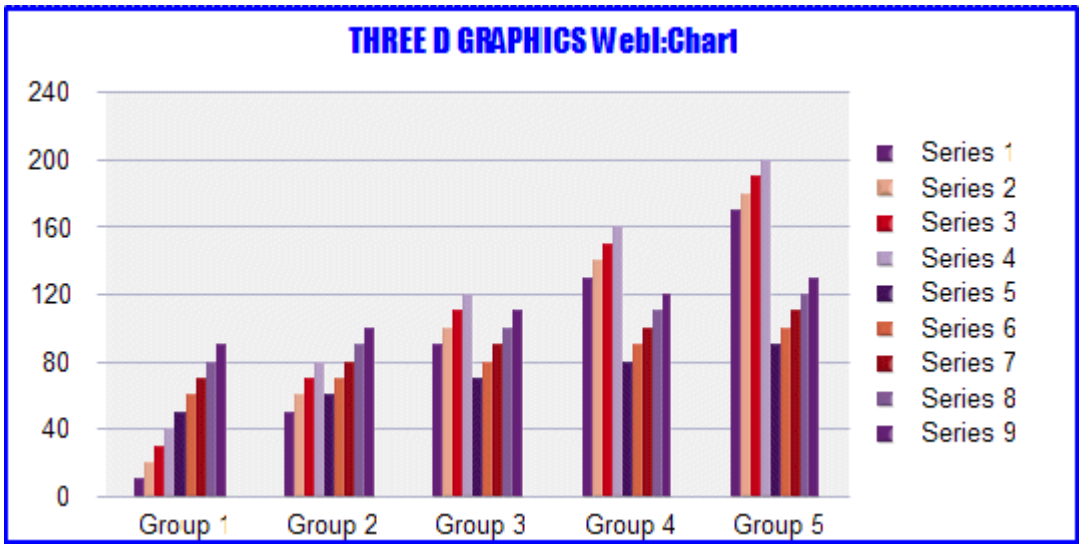
```
@RISER_BORDER bShow
```

PARAMETERS:

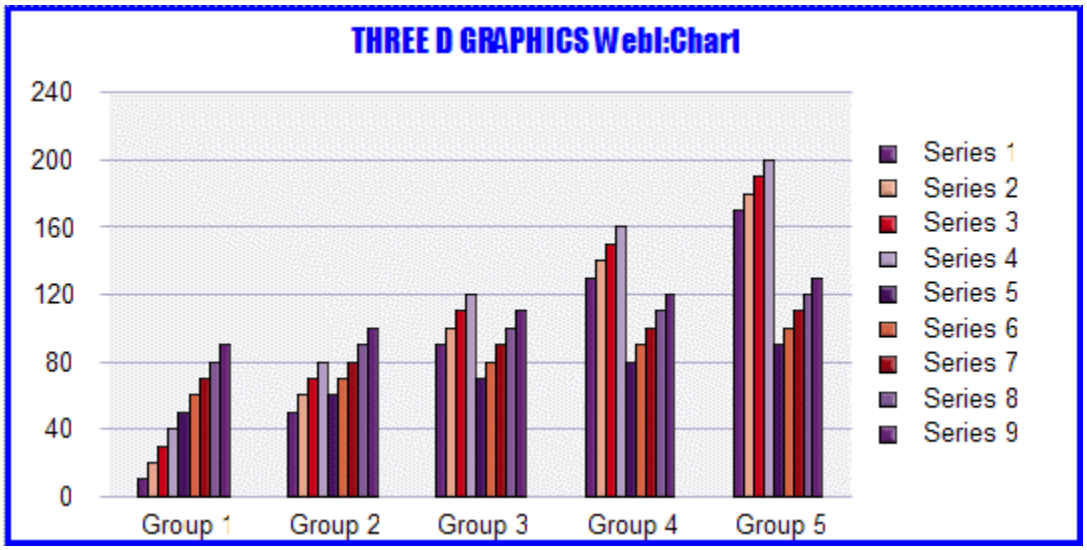
bShow; 0 = Turn OFF border line on risers/markers, 1= Turn ON border line on risers/markers.

EXAMPLE:

```
@RISER_BORDER 0
```




```
@RISER_BORDER 1
```



PERSISTENT:

YES

NOTE: This macro is not currently compatible with the 3D Look property. The 3D Look property should be disabled.

@RISER_OVERLAP (Riser Overlap)

This macro sets the amount of overlap between risers in a side-by-side/clustered bar chart.

SYNTAX:

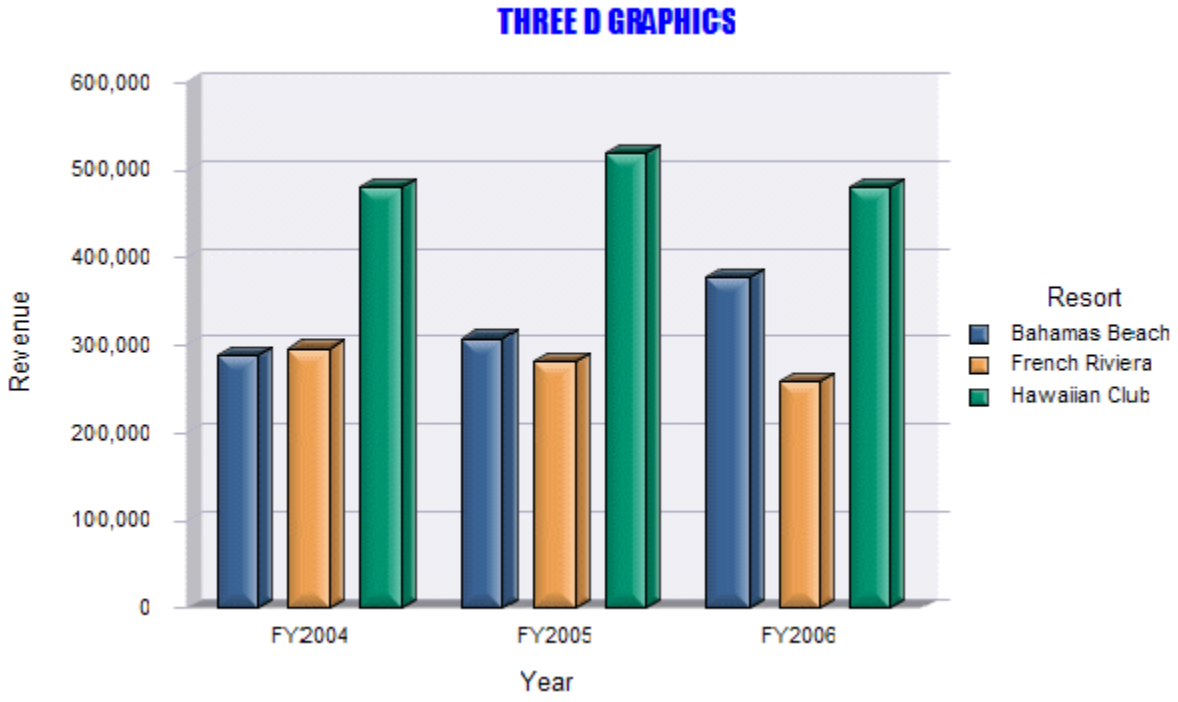
```
@RISER_OVERLAP nOverlap
```

PARAMETERS:

nOverlap; -100...100 selects the amount of overlap. Negative numbers make the risers overlap. A value of -100 will place all risers in a group draw on top of each other. Larger numbers will draw the risers farther apart.

EXAMPLE:

```
@RISER_OVERLAP 50
```



PERSISTENT:

YES

@RISER_WIDTH (Riser Width)

This macro can be used to change the width of risers in a bar chart.

SYNTAX:

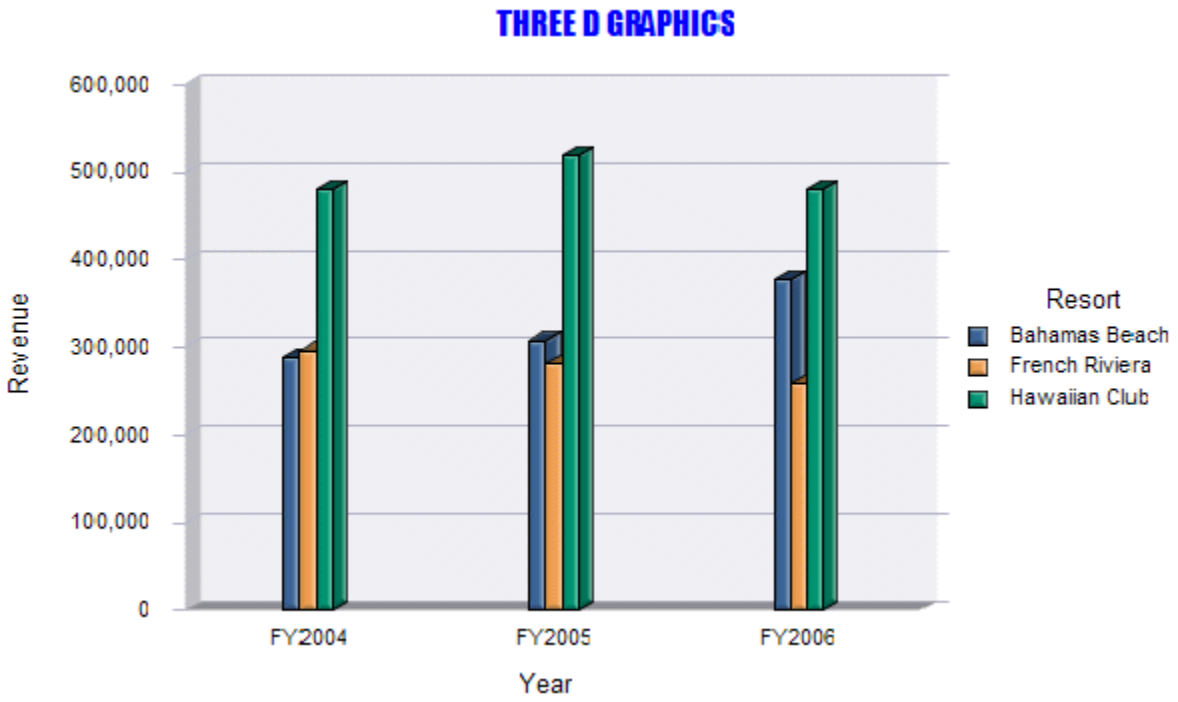
```
@RISER_WIDTH nWidth
```

PARAMETERS:

nWidth; 0...100 selects the width of risers.

EXAMPLE:

```
@RISER_WIDTH 20
```



PERSISTENT:

YES

@SZ (Size of Markers)

This macro sets the size of the markers in any chart type where a marker is drawn to represent a data point (i.e., Line charts, Scatter charts, Polar charts, Box Plots).

SYNTAX:

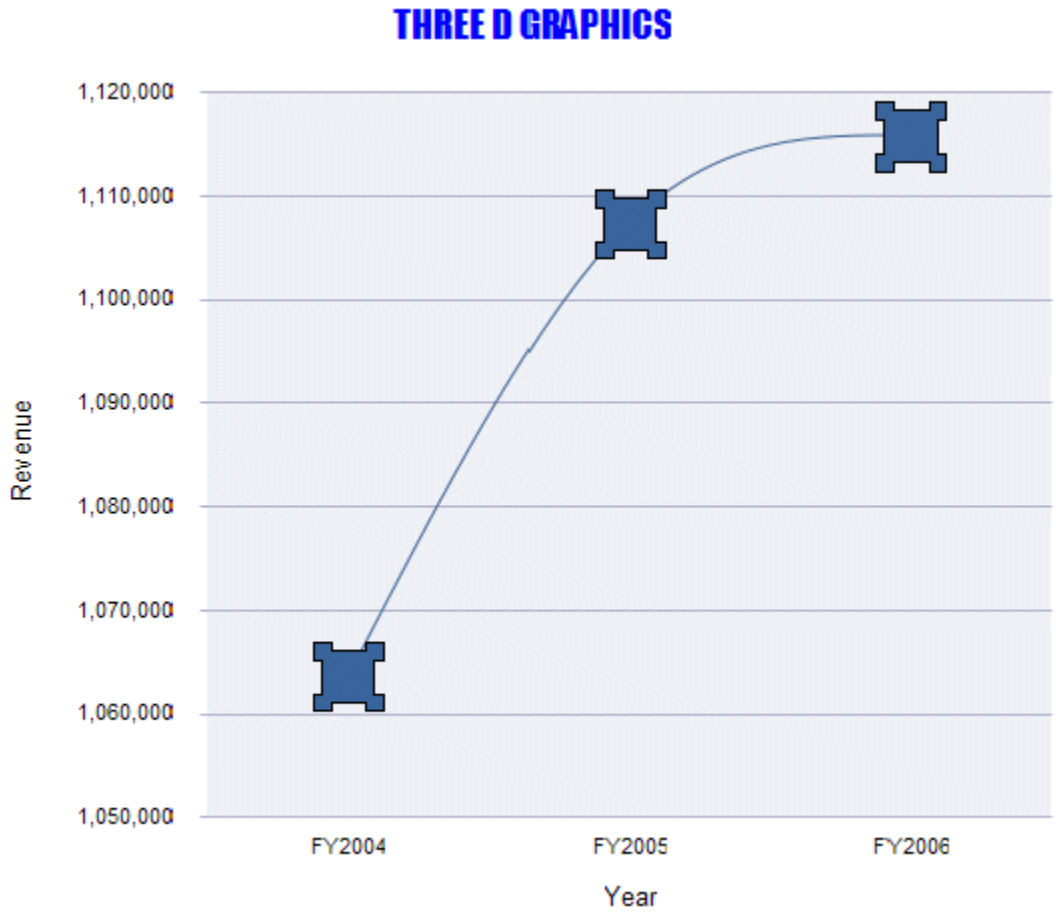
```
@SZ nValue
```

PARAMETERS:

nValue; Size of markers (1...100)

EXAMPLE:

```
@SZ 100
```



PERSISTENT:

NO

ALSO SEE:

@MS & @MC


WebI: CHART

Section 7: Data & Data Text

Use these macros to manage data and data text:

- @DATATEXT_COLORMODE; Data text color mode for Bubble Charts
- @DLT; Data Line Type
- @DP; Data Point Override
- @DPC; Data Point Clear
- @FORCE_DATATEXT_CURRENCY; Force data text to currency format
- @OFFSCALE_Y1; Define handling of off-scale Values on the Y1-Axis
- @RDT; Rotate Data Text
- @Y_ZERO; Include/Exclude zero to calculate Y1-axis auto-scale

@DATATEXT_COLORMODE (Data Text Color Mode for Bubble Charts)

This macro enables/disables special data text color mode in Bubble Charts.

SYNTAX:

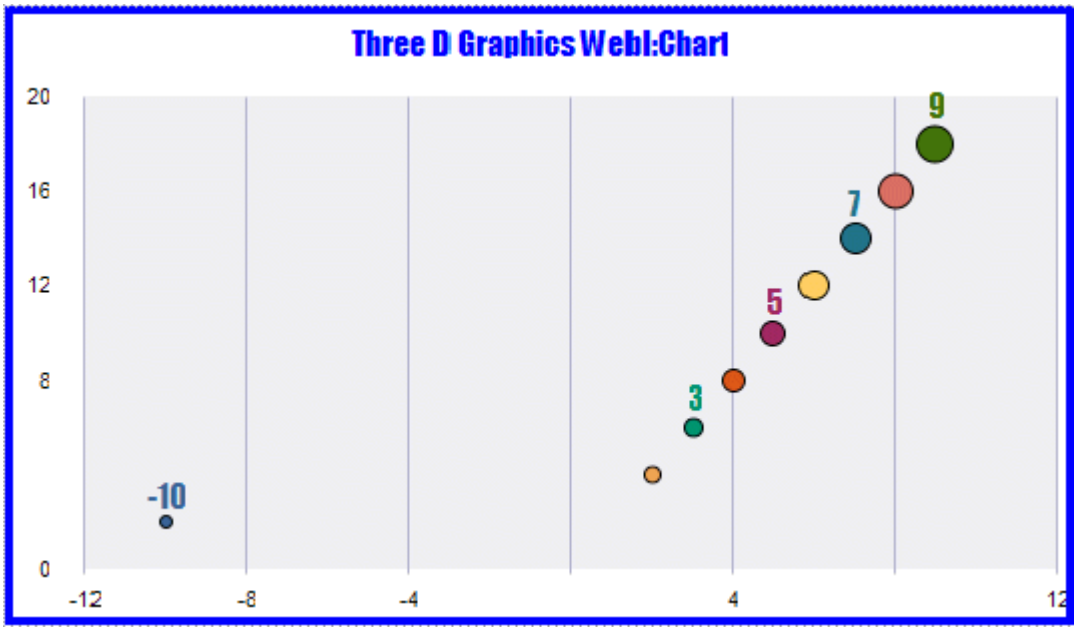
```
@DATATEXT_COLORMODE bColorMode
```

PARAMETERS:

bColorMode; 0/1
0 = Use normal (typically black) to color data text in the Bubble Chart
1 = Use the color of the bubble to draw the data text

EXAMPLE:

```
@DATATEXT_COLORMODE 1
```



PERSISTENT:

NO

@DLT (Data Line Type)

On Line and 2D-Scatter charts, this macro can be used to draw markers only, lines only, or markers and lines.

SYNTAX:

```
@DLT nSeries nType
```

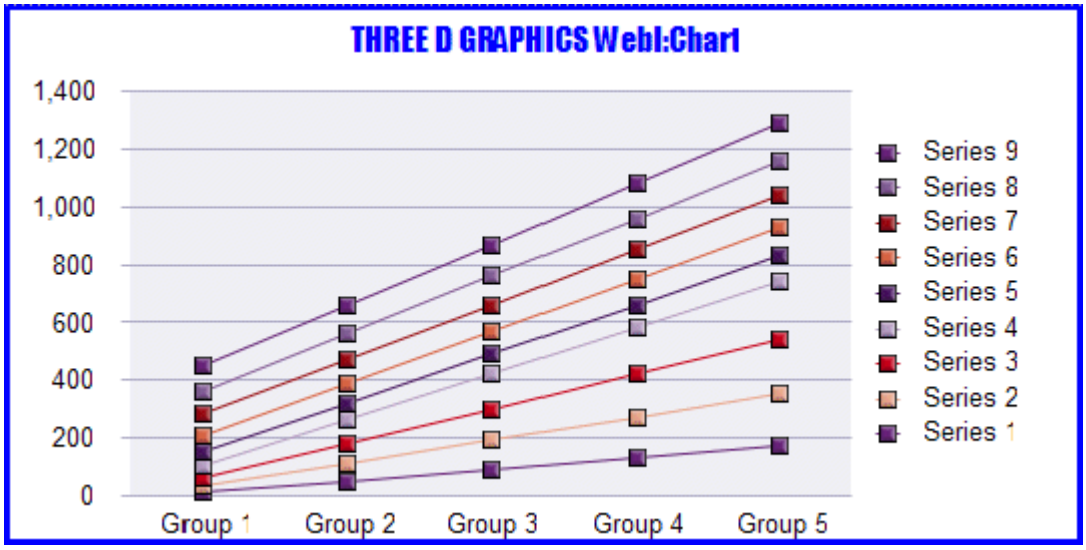
PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nType; 1 = draw markers only, 2 = draw lines only, 3 = draw markers and lines.

EXAMPLE:

```
@DLT -1 3
```



PERSISTENT:

YES

NOTES:

If you use @GRAPHTYPE to create a line chart or 2D scatter chart, this macro must be before the graph type selection.

@DP (Data Point Override)

This macro can be used to arbitrarily set a value for a bar, line, area, or pie chart by specifying a series, group and value.

SYNTAX:

```
@DP nSeries nGroup fValue
```

PARAMETERS:

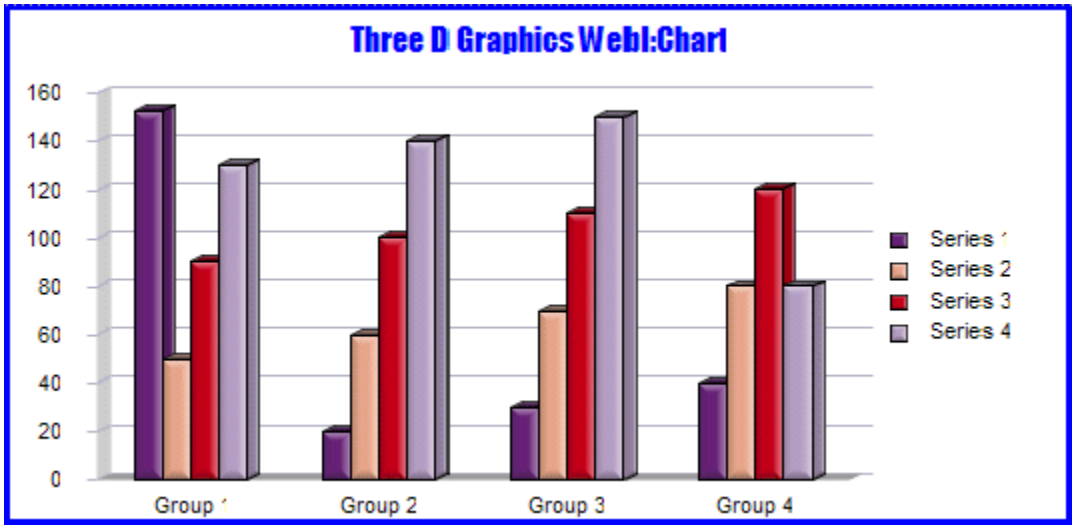
nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nGroup; -1...*n* (where: *n* = the total number of groups in the chart).
-1 = apply to all groups, 0 = Group 1, 1 = Group 2, etc.

fValue; The REAL value to be assigned to *nSeries*/*nGroup*.

EXAMPLE:

```
@DP 0 0 152
```



PERSISTENT:

NO

ALSO SEE:

@DPC

@DPC (Data Point Clear)

This macro can be used to arbitrarily CLEAR a value (i.e., set to NULL) for a specified series and group in a bar, line, area, or pie chart.

SYNTAX:

```
@DPC nSeries nGroup
```

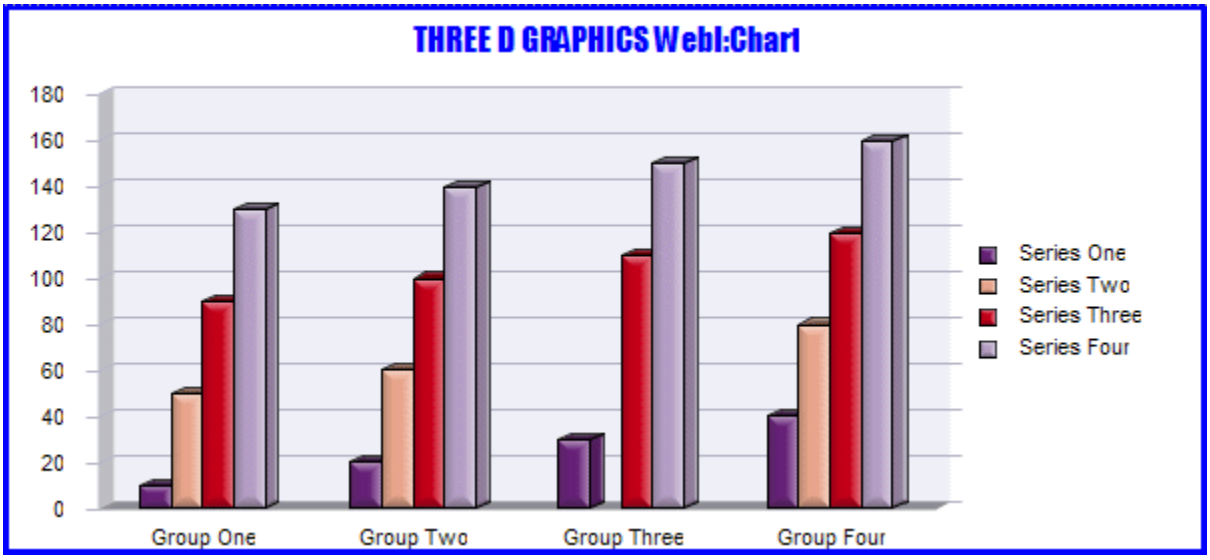
PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nGroup; -1...*n* (where: *n* = the total number of groups in the chart).
-1 = apply to all groups, 0 = Group 1, 1 = Group 2, etc.

EXAMPLE:

```
@DPC 2 1
```



PERSISTENT:

NO

ALSO SEE:

@DP

@FORCE_DATATEXT_CURRENCY (Force Data Text to Currency Format)

This macro forces data text numbers to be formatted with a given currency and precision.

SYNTAX:

```
@FORCE_DATATEXT_CURRENCY nCurrency nPrecision
```

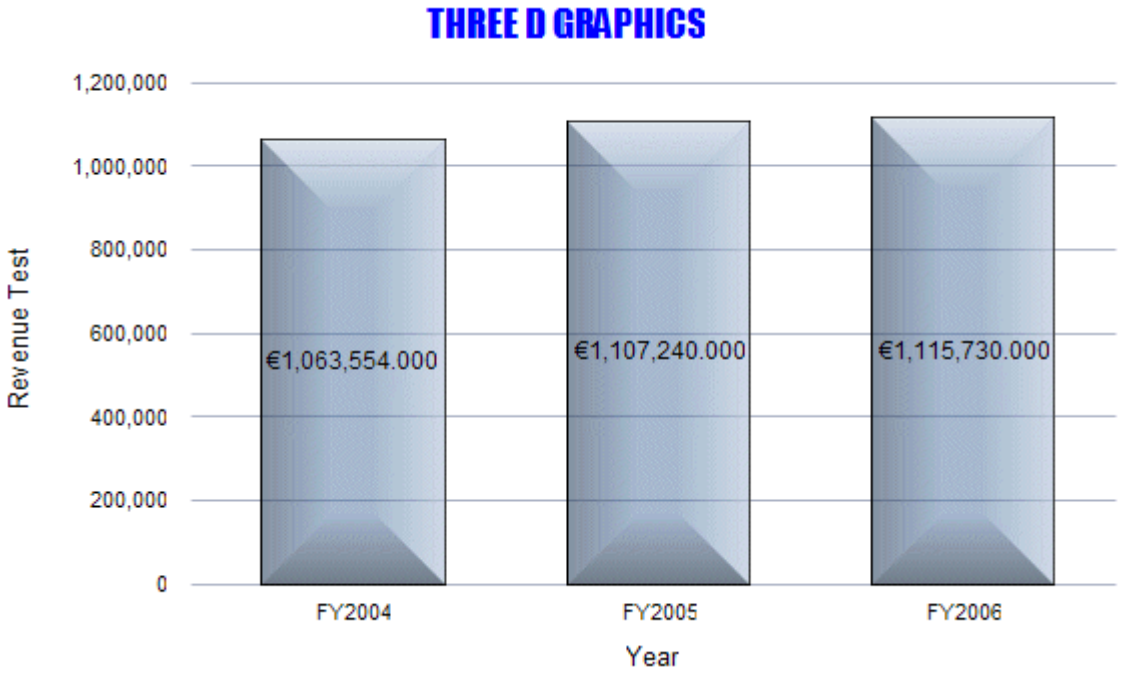
PARAMETERS:

nCurrency; 0...3 selects a currency format.
0=US Dollars
1=British Pound
2=Japanese Yen
3=Euro

nPrecision; 0...3 = Number of decimal places

EXAMPLE:

```
@FORCE_DATATEXT_CURRENCY 3 3
```



PERSISTENT:

YES

@OFFSCALE_Y1 (Y1-Axis Off-Scale Values)

This macro can be used to specify how off-scale values are charted on the Y1-axis.

SYNTAX:

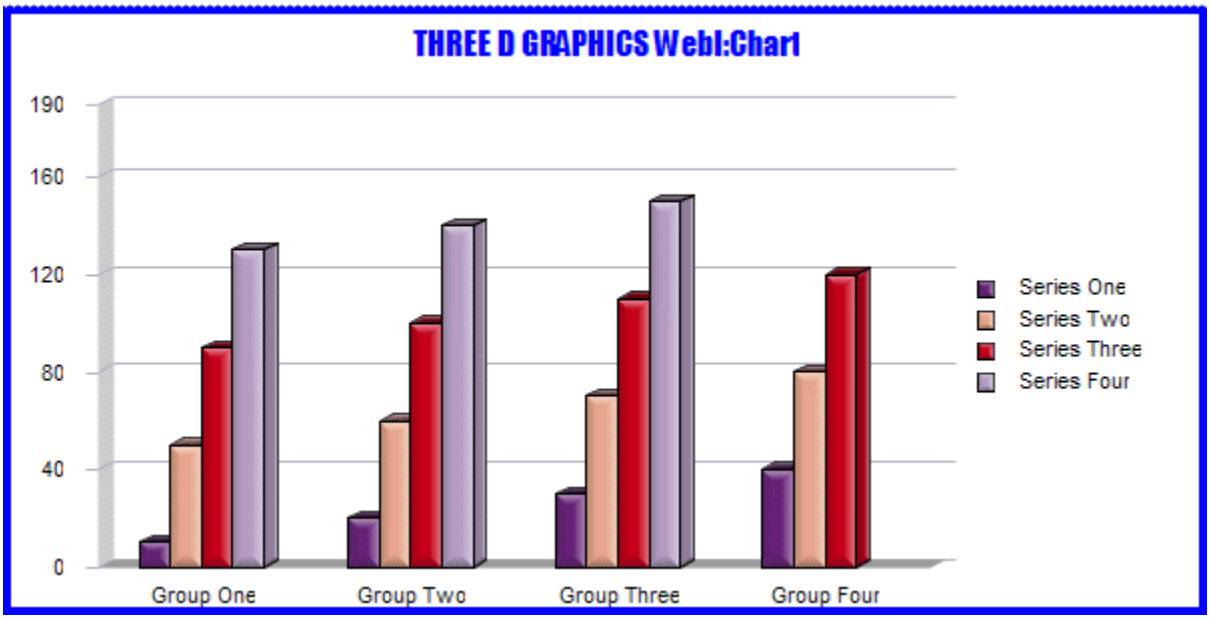
```
@OFFSCALE_Y1 nOffScaleMode
```

PARAMETERS:

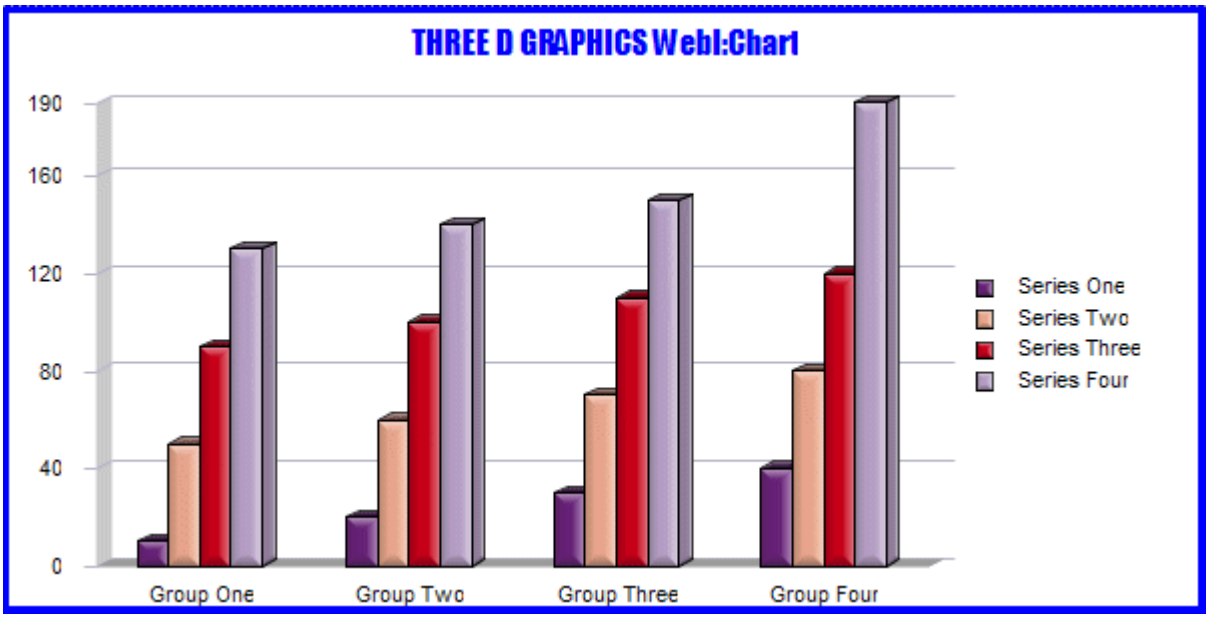
nOffScaleMode; 0...2. 0=Do not show off-scale values. 1=Graph off-scale values at scale maximum. 2=Clip off-scale values against chart frame

EXAMPLE:

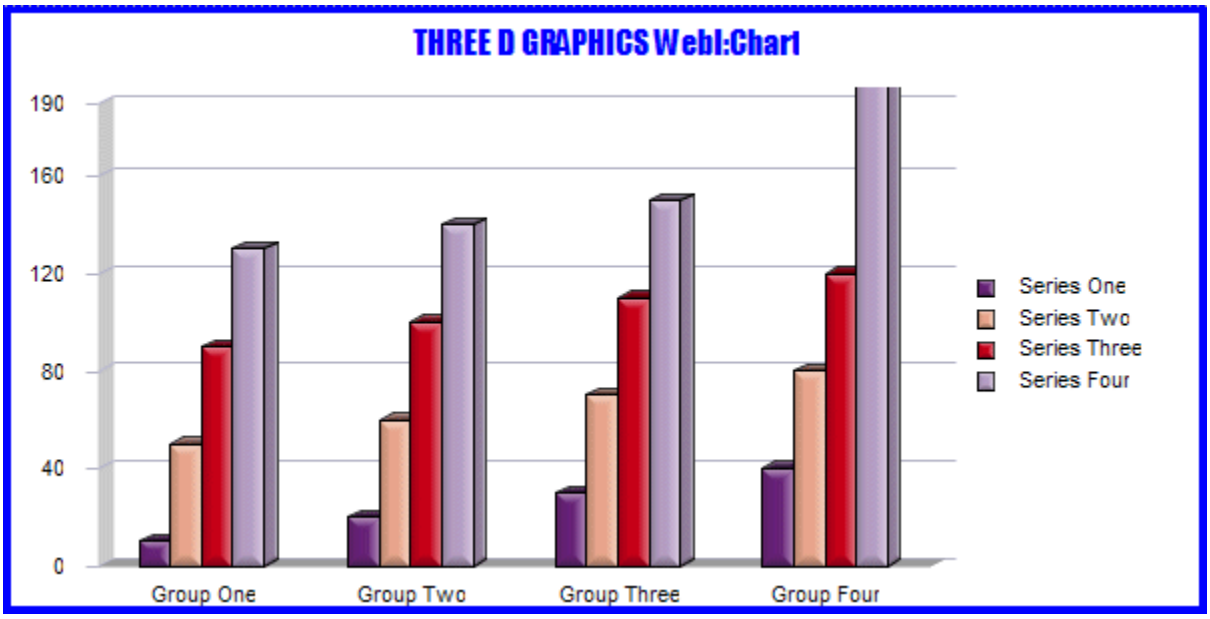
```
@OFFSCALE_Y1 0
```




```
@OFFSCALE_Y1 1
```




```
@OFFSCALE_Y1 2
```



PERSISTENT:

NO

@RDT (Rotate Data Text)

This macro can be used to rotate data text or to draw data text in hotel mode (vertically).

SYNTAX:

```
@RDT nRotateMode
```

PARAMTERS:

nRotateMode; 0...6 selects one of the following:

0 = No rotation

2 = Rotate data text 90 degrees

3 = Rotate data text 180 degrees

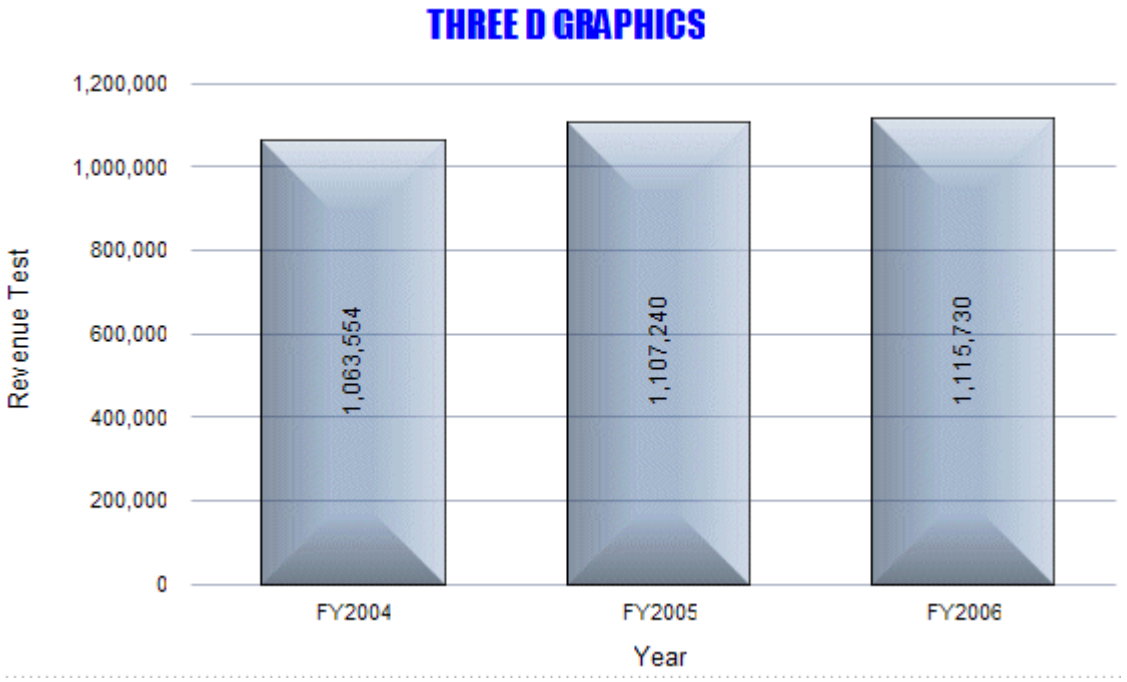
4 = Rotate data text 270 degrees

5 = Rotate data text 45 degrees

6 = Rotate data text 315 degrees

EXAMPLE:

```
@RDT 2
```



PERSISTENT:

YES

@Y_ZERO (Include/Exclude Zero in Y1-Auto-Scale)

This macro includes/excludes zero in the calculation of Y1-Axis auto-scale.

SYNTAX:

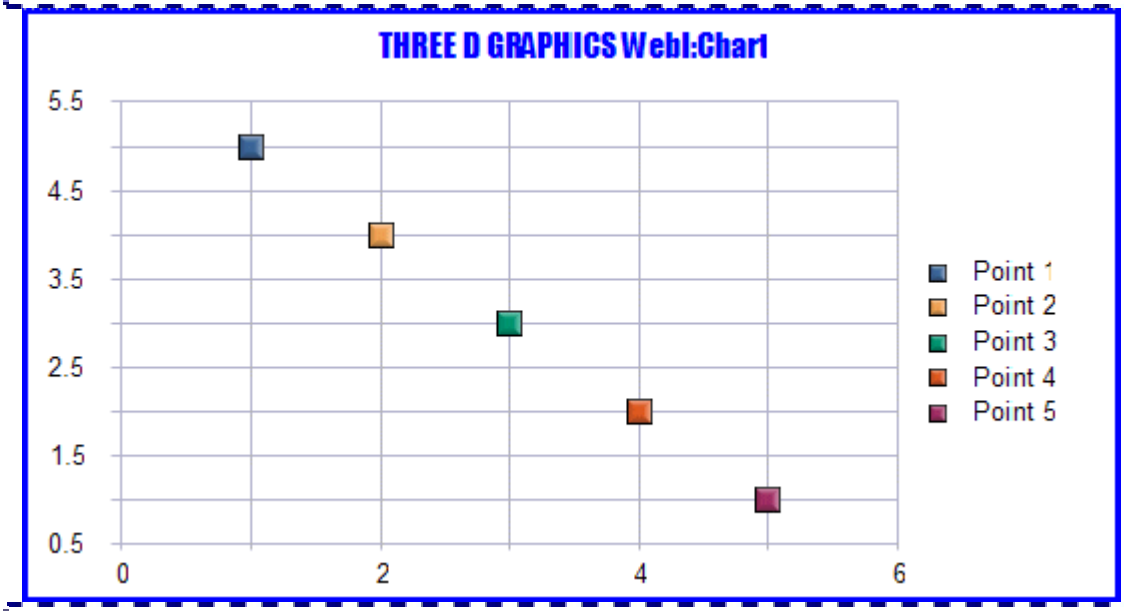
```
@Y_ZERO bExcludeZero
```

PARAMETERS:

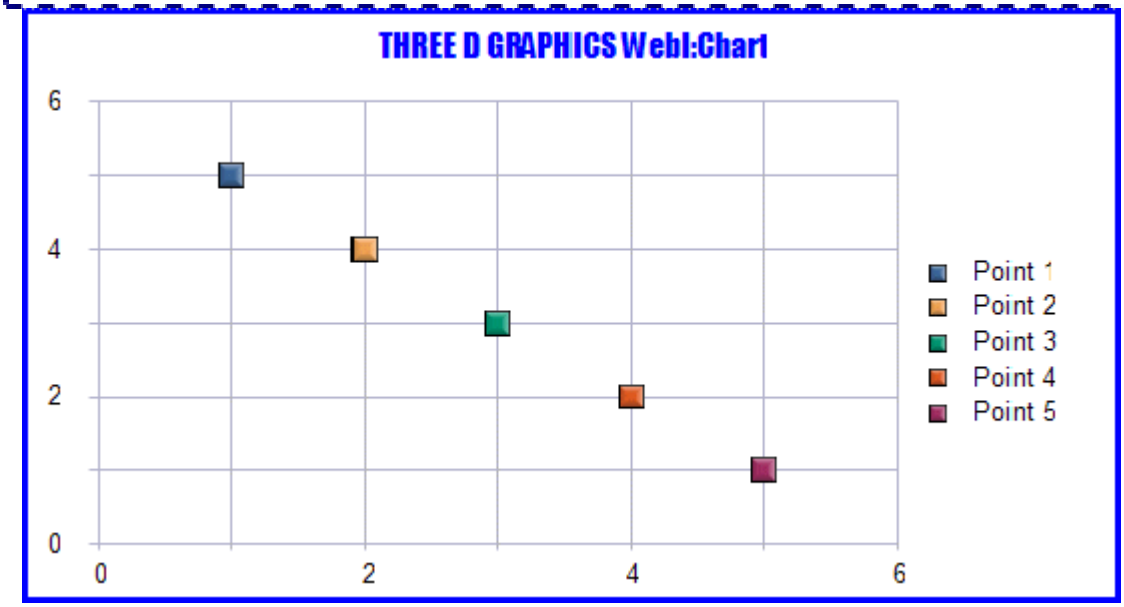
bExcludeZero; 1 = Zero is NOT used in calculating Y1-axis auto-scale. 0=Zero IS used in calculating Y1-axis auto-scale

EXAMPLE:

```
@Y_ZERO 1
```




```
@Y_ZERO 0
```



PERSISTENT:

NO

Section 8: Color & Special Effects

The following macros can be used to apply color and special effects to a chart:

- @ALPHA; Apply Alpha Channel Transparency to a riser or marker
- @AUTO_COLOR; Activate Automatic Color Mode
- @BEVEL; Draw a bevel effect on a chart object
- @COLOR_MODE; Choose a Color Mode (Color by Series or Color by Group)
- @COLOR_SCHEME; Activate a pre-defined color scheme
- @COND_COLOR & @COND_COLOR2; Apply color to risers (bar/line/area) based on conditions
- @CURVED_LINES; Apply curved lines to an line or area chart
- @DEFINE_SCHEME; Define Color Scheme
- @GCOLOR; Color a Chart Object
- @GRAD; Gradient Definition
- @GRAD_ABA; Mirrored Gradient Definition
- @HQ; Enable/Disable high quality rendering
- @PICT; Picture Definition
- @PICTF; Scaled Picture Definition
- @SHADOW; Apply a Drop Shadow to a chart object

Also see the @MC macro in "Box Plot Macros" to define markers colors in Box Plots.

Also see the @MCOLOR and @PAT macros in "Risers & Markers" to apply a color or a pattern to risers and markers.

@ALPHA (Alpha Channel Transparency)

This macro sets the Alpha Channel Transparency of markers and risers on a chart. The *nValue* parameter selects the amount of opaqueness/transparency.

SYNTAX:

```
@ALPHA nSeries nValue
```

PARAMETERS:

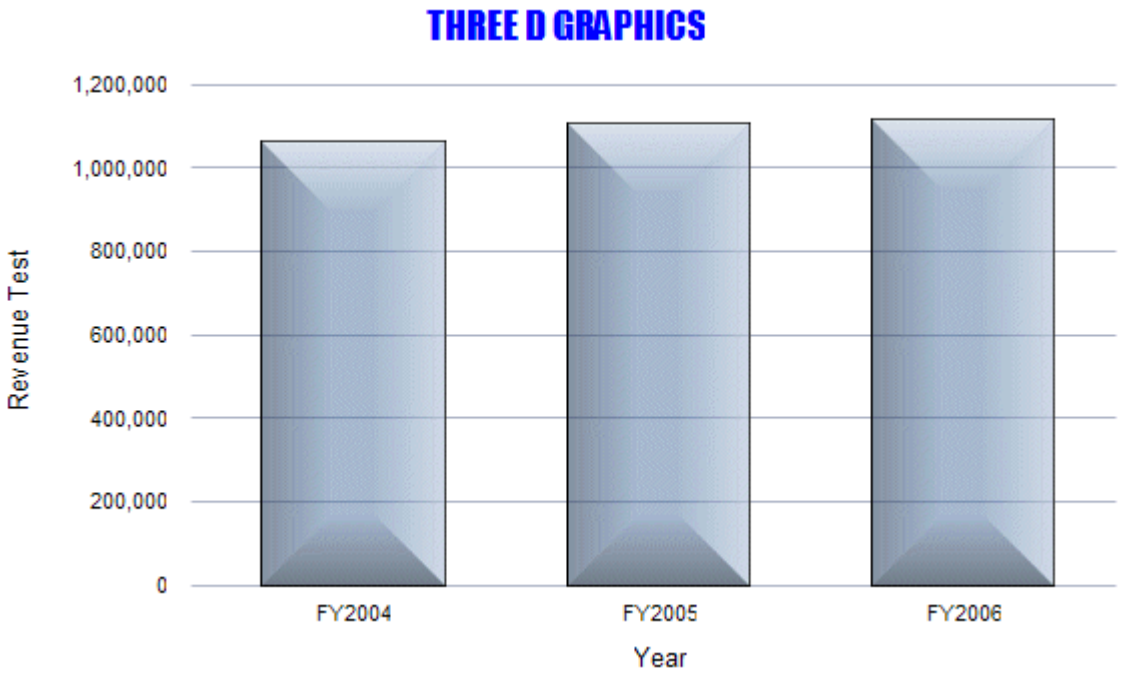
nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nValue; 0...255 selects the transparent level.
255 (the default) = no transparency.
0 = fully transparent.

EXAMPLE:

The example creates a chart with a transparency level of 100 on series 1 and a level of 150 on series 2.

```
@ALPHA -1 112
```



PERSISTENT:

YES

@AUTO_COLOR (Automatic Color Mode)

This macro enables/disables automatic color mode. When a single series appears in a BAR chart and this macro is enabled, the chart is automatically set to color-by-group.

SYNTAX:

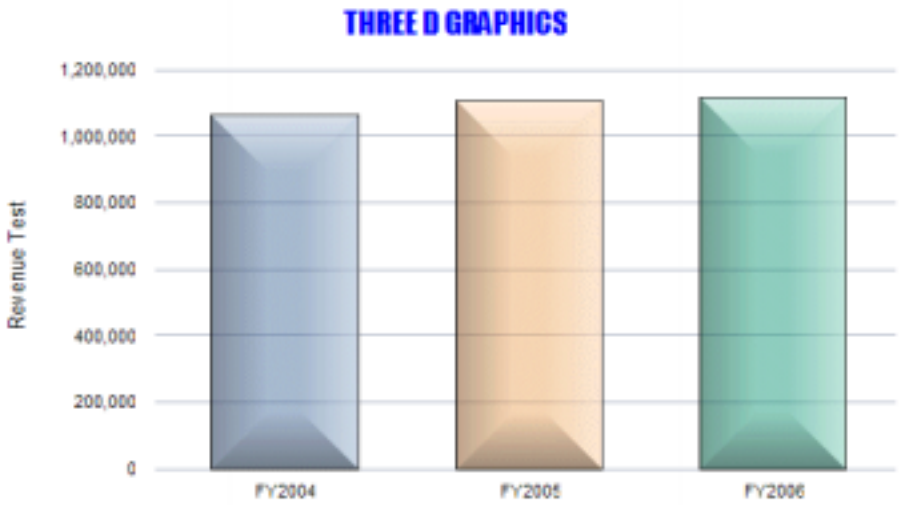
```
@AUTO_COLOR bActivate
```

PARAMETERS:

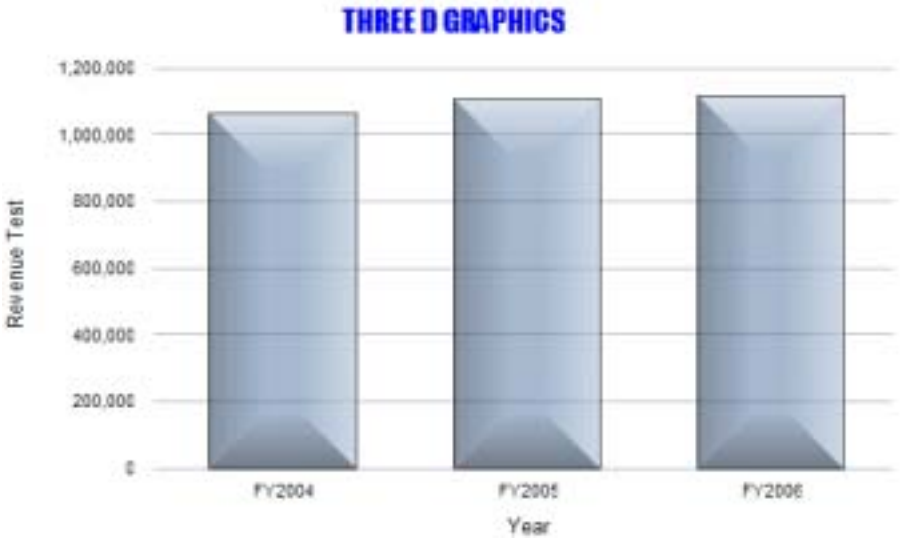
bActivate; 0=Disable Auto-Color Mode, 1=Enable Auto-Color Mode

EXAMPLE:

```
@AUTO_COLOR 1
```




```
@AUTO_COLOR 0
```



PERSISTENT:

YES

@BEVEL (Bevel Chart Object)

This macro can be used to apply a bevel effect to a chart object.

SYNTAX:

```
@BEVEL nObject nBevelType
```

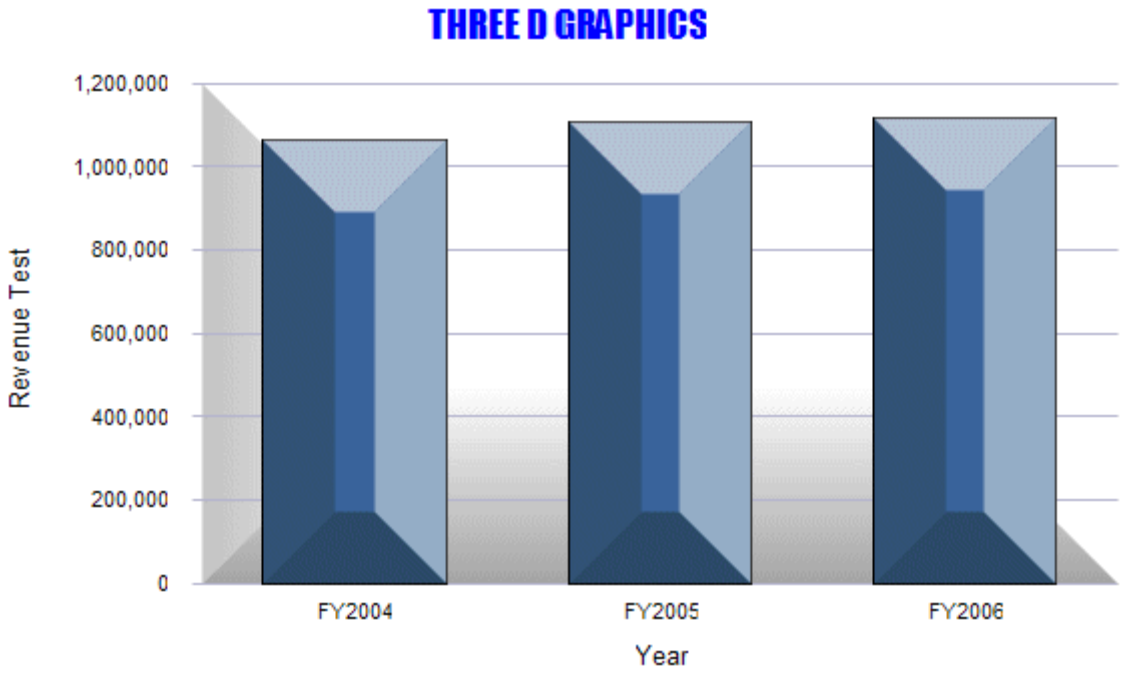
PARAMETERS:

nObject; 1...3. 1=Chart Background, 2=Chart Frame. 3=Risers

nBevelType; 0...3. 0=None, 1=Smooth Edge, 2=Chisel Edge, 3=Donut. Bevel types 1 and 2 can only be applied to rectangle and polygons objects. Bevel type 3 can only be applied to ellipses and slices.

EXAMPLE:

```
@BEVEL 3 2
```



PERSISTENT:

YES (You can use *nBevelType* zero (0) to remove the bevel)

@COLOR_MODE (Color Mode)

This macro sets the color mode (Color by Series or Color by Group) to be used in a chart.

SYNTAX:

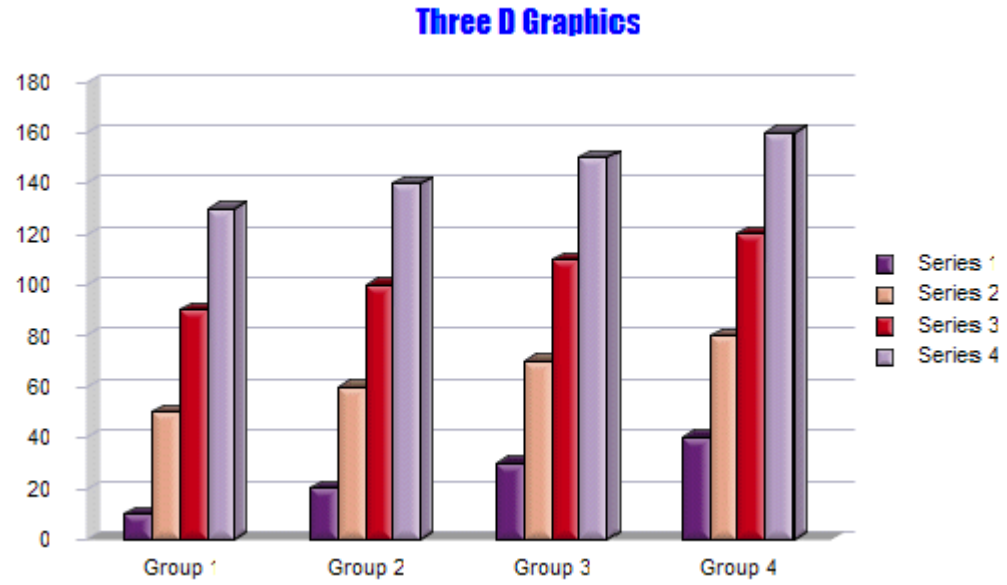
```
@COLOR_MODE bColorMode
```

PARAMETERS:

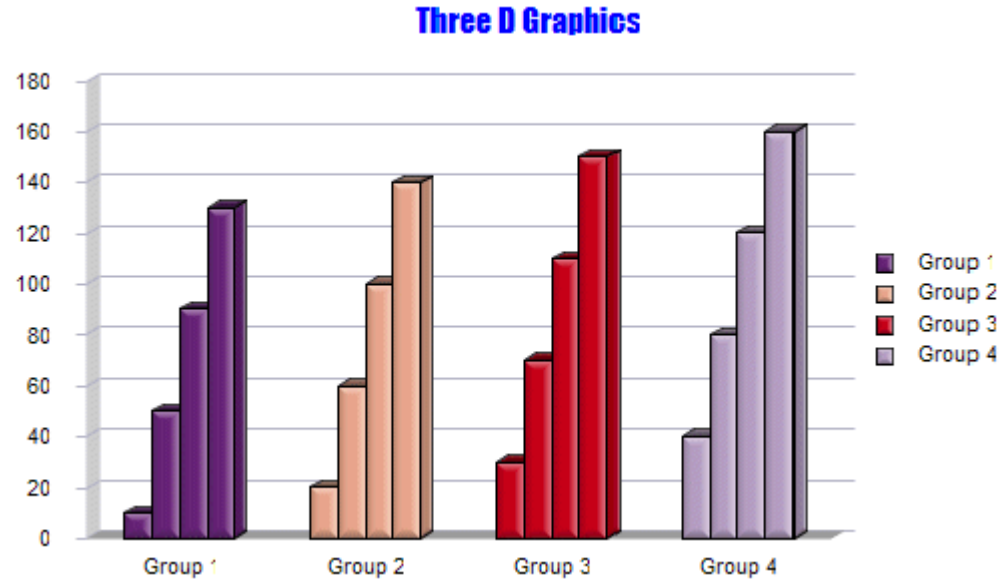
bColorMode; 0 = Color By Series, 1 = Color By Group

EXAMPLE:

```
@COLOR_MODE 0
```




```
@COLOR_MODE 1
```



PERSISTENT:

NO

@COLOR_SCHEME (Activate a Color Scheme)

This macro activates one of six preset color schemes that will be used to color chart risers/markers. Each color scheme consists of a group of eight colors that is repeated after the first eight series.

SYNTAX:

```
@COLOR_SCHEME nScheme
```

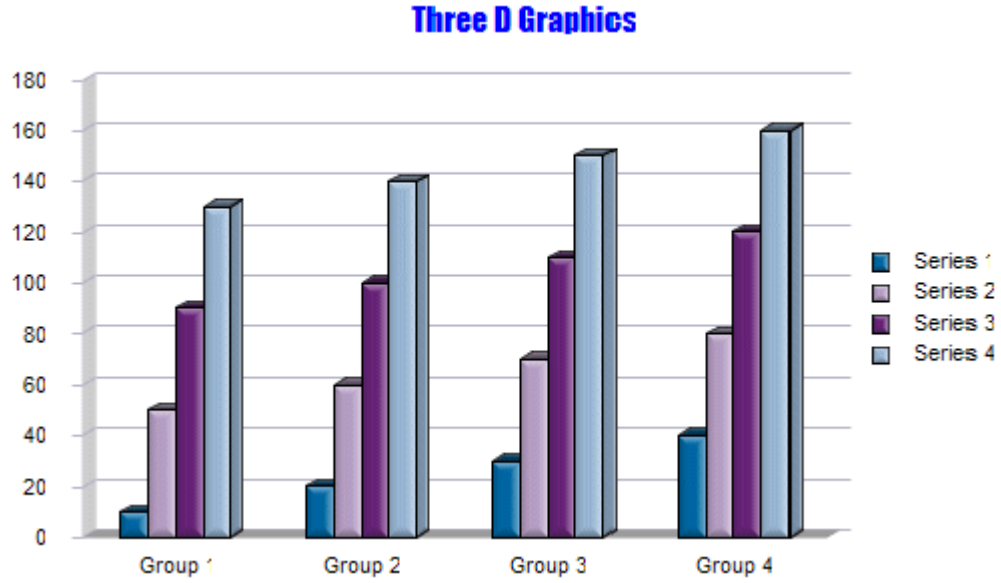
PARAMETERS:

nScheme; 0...5 selects one of the following color schemes

nScheme	Color Scheme
0	
1	
2	
3	
4	
5	

EXAMPLE:

```
@COLOR_SCHEME 4
```



PERSISTENT:

YES

ALSO SEE:

@DEFINE_SCHEME

@COND_COLOR & @COND_COLOR2 (Conditional Colors)

These macros create a “conditional color” that will be applied to a marker/riser when a specified condition is matched.

SYNTAX:

```
@COND_COLOR nSeries nGroup nCondition fValue nRed nGreen nBlue
@COND_COLOR2 nSeries nGroup nCondition fValue nRed nGreen nBlue
```

PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart).
-1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nGroup; -1...*n* (where: *n* = the total number of groups in the chart).
-1 = apply to all groups, 0 = Group 1, 1 = Group 2, etc.

nCondition; 0...7 selects one of the following conditions:

0 = if the value of a bar/line/area marker is LESS THAN *fValue,* the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used.

1 = if the value of a bar/line/area marker is LESS THAN OR EQUAL TO *fValue,* the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used.

2 = if the value of a bar/line/area marker is GREATER THAN *fValue,* the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used.

3 = if the value of a bar/line/area marker is GREATER THAN OR EQUAL TO *fValue,* the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used.

4 = if the value of a bar/line/area marker is EQUAL TO *fValue,* the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used.

5 = if the value of a bar/line/area marker is NOT EQUAL TO *fValue,* the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used.

6 = if the current group is GREATER THAN OR EQUAL TO *nGroup*, the Conditional color specified by *nRed*/*nGreen*/*nBlue* will be used. This can be used to change the color of the riser based on the Group ID alone which is useful for something like a projection. See Example. When this condition is used, the *fValue* parameter is ignored.

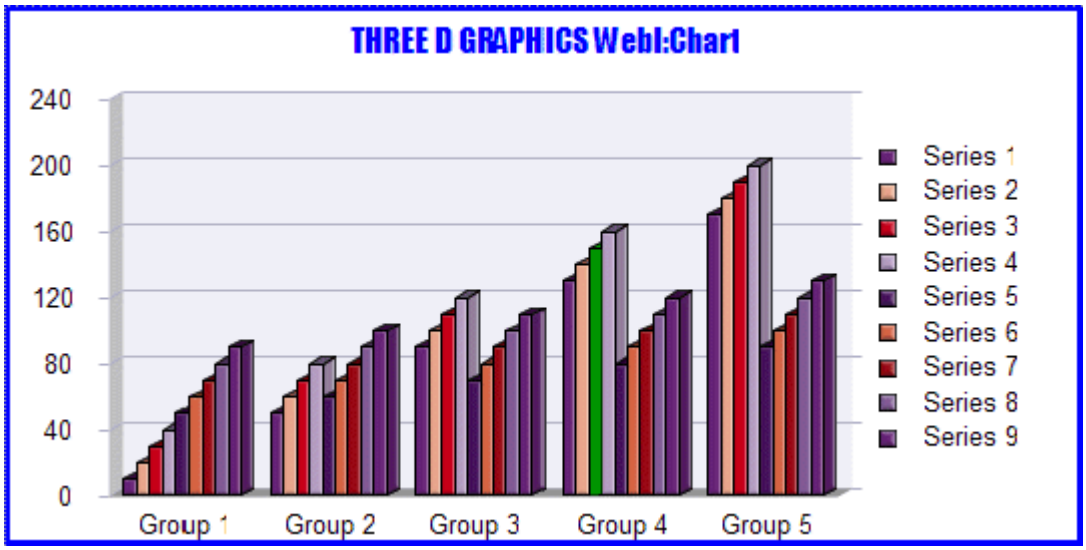
7 = force all series in *nGroup* to the conditional color regardless of *fValue*.

fValue; Value to compare the riser value to in order to determine whether or not to apply the Conditional color

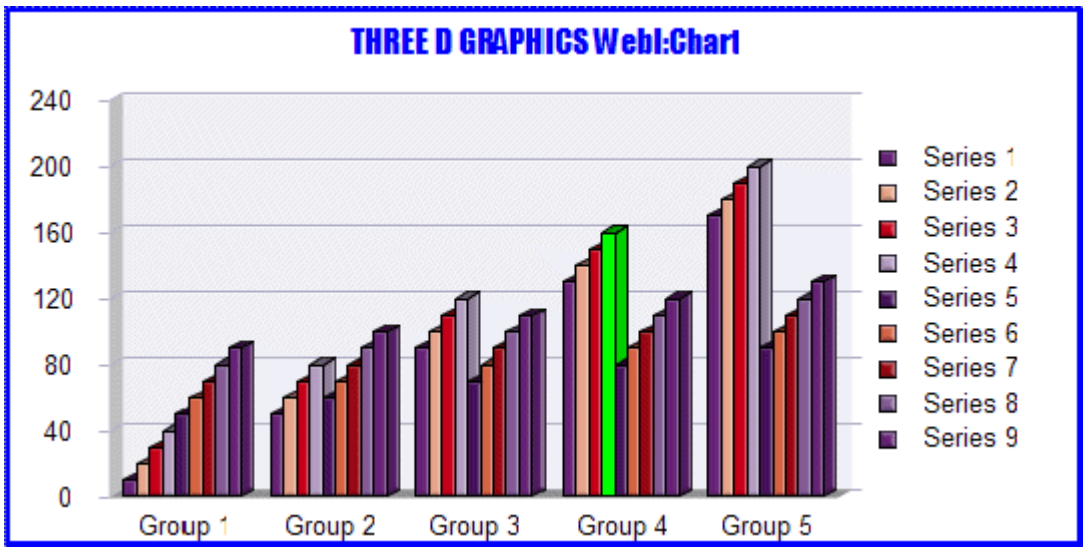
nRed, nGreen, nBlue; 0...255 specifies the RGB value of the Conditional color.

EXAMPLE:

```
@COND_COLOR 2 3 0 200 0 255 0
```




```
@COND_COLOR 3 3 1 160 0 255 0
```



PERSISTENT:

NO

NOTES:

You can set two conditional colors per chart using @COND_COLOR and @COND_COLOR2. However, you can also use the *nSeries* and *nGroup* settings to apply to ALL RISERS (*nSeries* = -1, *nGroup* = -1), ALL GROUPS IN A SERIES (*nSeries* = 0...*n*, *nGroup* = -1), ALL SERIES IN A GROUP (*nSeries* = -1, *nGroup* = 0...*n*) or A PARTICULAR RISER (*nSeries* = 0...*n*, *nGroup* = 0...*n*).

@CURVED_LINES (Enable Curved Lines)

This macro enables/disables curved lines in a line or area chart.

SYNTAX:

```
@CURVED_LINES bEnable
```

PARAMETERS:

bEnable; 0=Disable Curved Lines, 1=Enable Curved Lines

EXAMPLE:

```
@CURVED_LINES 1
```

PERSISTENT:

YES

@DEFINE_SCHEME (Define Color Scheme)

This macro can be used to define an 8-series color scheme using Red-Green-Blue values that will be used to color markers/risers. An RGB value can be hex (0xFF0000=red) or decimal (16711680=red).

SYNTAX:

```
@DEFINE_SCHEME Color1 Color2 Color3 Color4 Color5 Color6 Color7
Color8
```

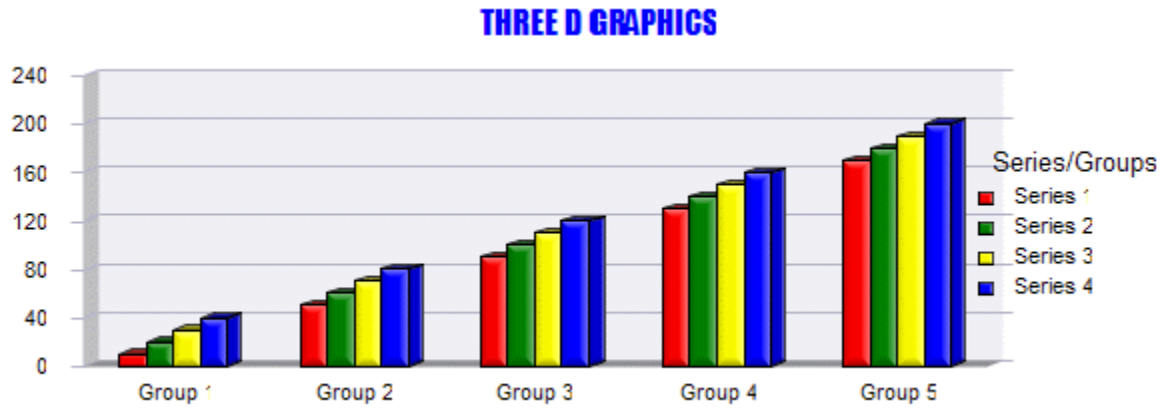
PARAMETERS:

Color1...Color8; Each color can defined in any of the following formats:

Data Type	Range	Example
Hex	0x000000...0xFFFFFF	0xFF0000 (Red)
HTML	#000000...#FFFFFF	#FF0000 = Red
Integer	three integers 0...255 specifying the red, blue, & green components of the color	255 0 0 = Red
String	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red

EXAMPLE:

```
THREE D GRAPHICS~ @FONTNAME 1 2 @SWAP 0 @HQ 1 @BEVEL 3 1
@DEFINE_SCHEME Red Green Yellow Blue Orange Grey Black Rose Green
```



PERSISTENT:

YES

ALSO SEE:

@COLOR_SCHEME

@GCOLOR (Color Chart Object)

This macro can be used to change the color of major objects in a chart.

SYNTAX:

```
@GCOLOR nObject ObjectColor
```

PARAMETERS:

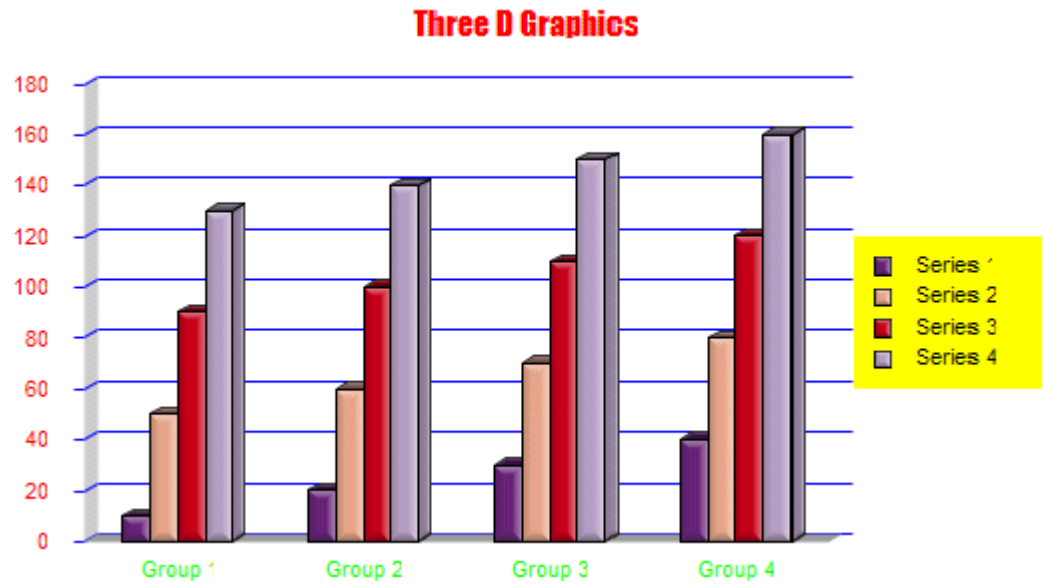
nObject; 0...14 selects one of the following chart objects:
0 = Chart Frame
1 = Legend Frame
2 = Title
3 = Reserved for Future Use
4 = Reserved for Future Use
5 = Y1 Axis Title
6 = Y2 Axis Title
7 = X Axis Title
8 = Y1 Axis Labels
9 = Y2 Axis Labels
10 = X Axis Labels
11 = Series Labels on Legend
12 = Y1 Major Gridlines
13 = Y2 Major Gridlines
14 = X1 or O1 Major Gridlines
15 = Gauge Face
16 = Gauge Border
18 = Gauge Needle Base

ObjectColor: The color of the object can be specified in one of the following formats:

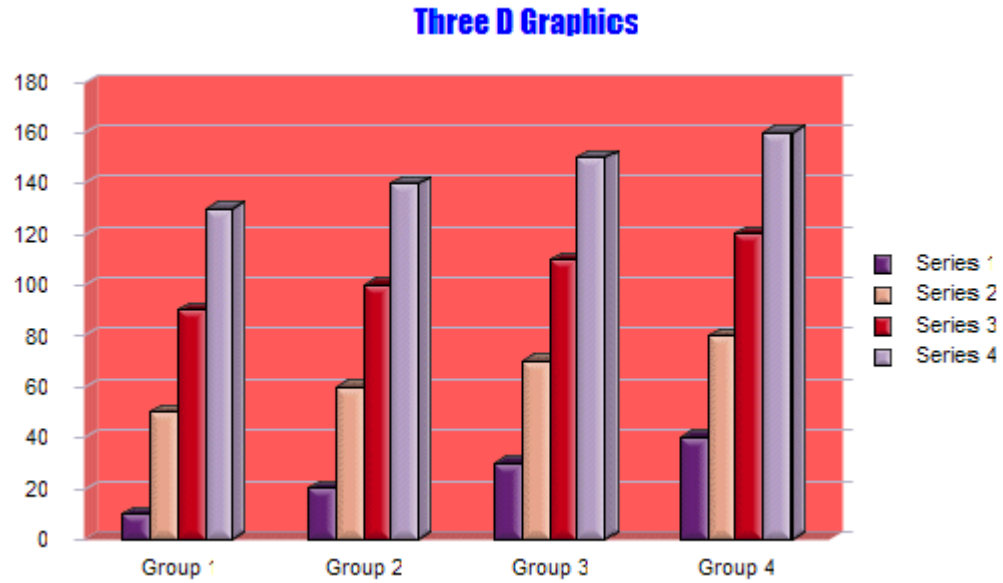
	Data Type	Range	Example
Color	Hex	0x000000...0xFFFFFF	0xFF0000 (Red)
	HTML	#000000...#FFFFFF	#FF0000 = Red
	Integer	three integers 0...255 specifying the red, blue, & green components of the color	255 0 0 = Red
	String	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red
Color + Transparency	String: Integer	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red:224 (almost opaque)
	HTML	#00000000...#FFFFFFE0 where: last two characters specify transparency level	#ff0000e0 #ff0000=Red e0=Transparency
Gradient	String	For area objects (0, 1, 11, 15, 16, 19) only, a gradient name defined by the @GRAD or @ GRAD_ABA macro	@GCOLOR 0 MyGradientDef
Picture	String	For area objects (0, 1, 11, 15, 16, 19) only, a gradient name defined by the @PICT or @PICTF macro	@GCOLOR 0 MyPictureDef

EXAMPLE:

```
@GCOLOR 1 255 255 0 @GCOLOR 2 255 0 0 @GCOLOR 8 255 0 0
@GCOLOR 12 0 0 255 @GCOLOR 10 0 255 0
```




```
@GCOLOR 0 #ff0000a0
```



PERSISTENT:

YES

@GRAD (Gradient Definition)

This macro can be used to define a gradient that can be applied to chart area objects.

SYNTAX:

```
@GRAD name angle startColor endColor
```

PARAMETERS:

name; Gradient name that can be used by other macros to assign the gradient to a chart object (e.g., @GCOLOR 0 MyGradient).

angle; 0...359 or "OVAL"

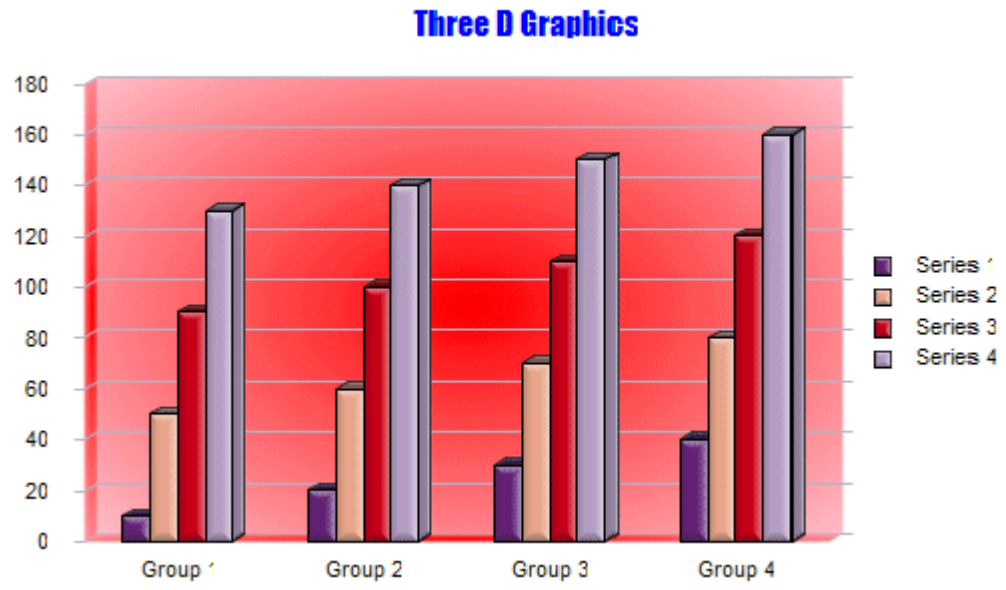
startColor; Start Color defined in one of the following formats.

endColor; End Color defined in one of the following formats.

	Data Type	Range	Example
Color	Hex	0x000000...0xFFFFFF	0xFF0000 (Red)
	HTML	#000000...#FFFFFF	#FF0000 = Red
	Integer	three integers 0...255 specifying the red, blue, & green components of the color	255 0 0 = Red
	String	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red
Color + Transparency	String: Integer	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red:224 (almost opaque)
	HTML	#00000000...#FFFFFFE0 where: last two characters specify transparency level	#ff0000e0 #ff0000=Red e0=Transparency

EXAMPLE:

```
@GRAD MyGradient OVAL Red Pink
@GCOLOR 0 MyGradient
```



@GRAD_ABA (Mirrored Gradient Definition)

This macro can be used to define a mirrored gradient that can be applied to chart area objects.

SYNTAX:

```
@GRAD_ABA name angle startColor endColor
```

PARAMETERS:

name; Gradient name that can be used by other macros to assign the gradient to a chart object (e.g., @GCOLOR 0 MyGradient).

angle; 0...359 or "OVAL"

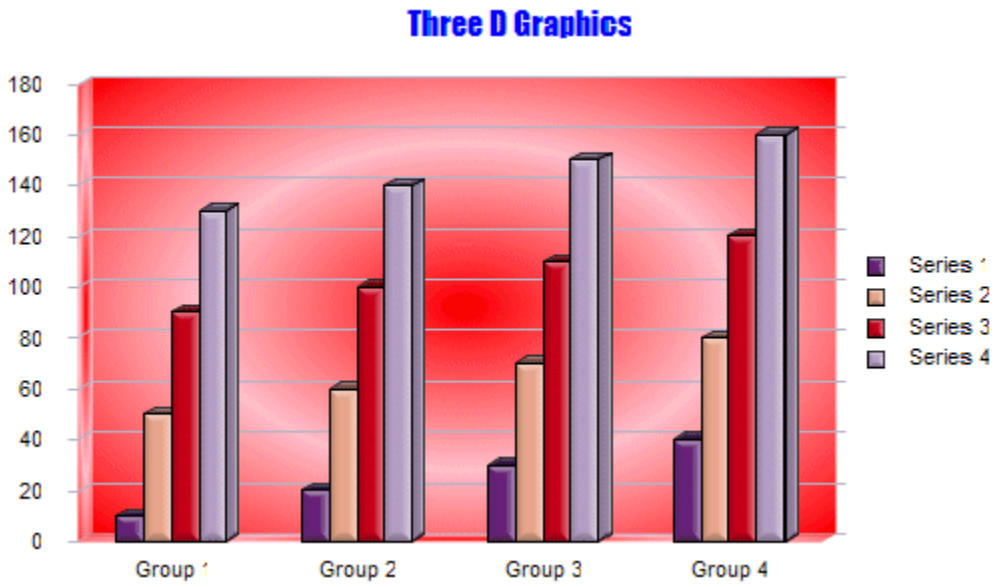
startColor; Start Color defined in one of the following formats.

endColor; End Color defined in one of the following formats.

	Data Type	Range	Example
Color	Hex	0x000000...0xFFFFFF	0xFF0000 (Red)
	HTML	#000000...#FFFFFF	#FF0000 = Red
	Integer	three integers 0...255 specifying the red, blue, & green components of the color	255 0 0 = Red
	String	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red
Color + Transparency	String: Integer	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red:224 (almost opaque)
	HTML	#00000000...#FFFFFFE0 where: last two characters specify transparency level	#ff0000e0 #ff0000=Red e0=Transparency

EXAMPLE:

```
@GRAD_ABA MyABAGradient OVAL Red Pink
@GCOLOR 0 MyABAGradient
```



@HQ (Enable High-Quality Rendering)

This macro enables/disables high-quality rendering.

SYNTAX:

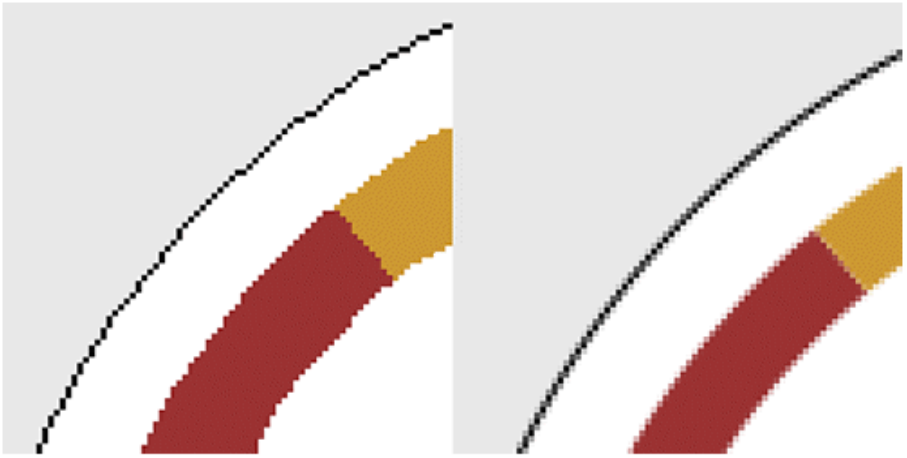
```
@HQ bEnable
```

PARAMETERS:

bEnable; 0/1
0=Disable High-Quality Rendering (i.e., use the standard drawing engine)
1=Enable High-Quality Rendering

EXAMPLE:

Close-Up View of High-Quality Rendering



@HQ 0 @HQ 1

PERSISTENT:

NO

@PICT (Picture Definition)

This macro can be used to define an image that can be applied to chart area objects.

SYNTAX:

```
@PICT name file
```

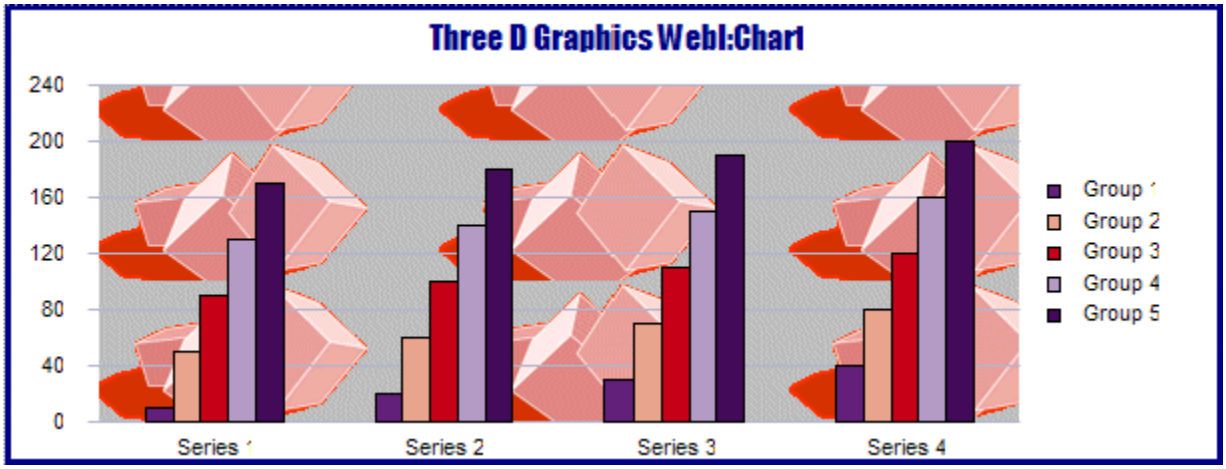
PARAMETERS:

name; Picture name that can be used by other macros to assign the picture to a chart object (e.g., @GCOLOR 0 MyPicture).

file; image file path (absolute or relative to working directory)

EXAMPLE:

```
@PICT pic1 c:\source\website_bezier\tdg\images\boxes_financial.gif
@GCOLOR 0 pic1
```



@PICTF (Scaled Picture Definition)

This macro can be used to define a scaled image that can be applied to chart area objects.

SYNTAX:

```
@PICTF name file
```

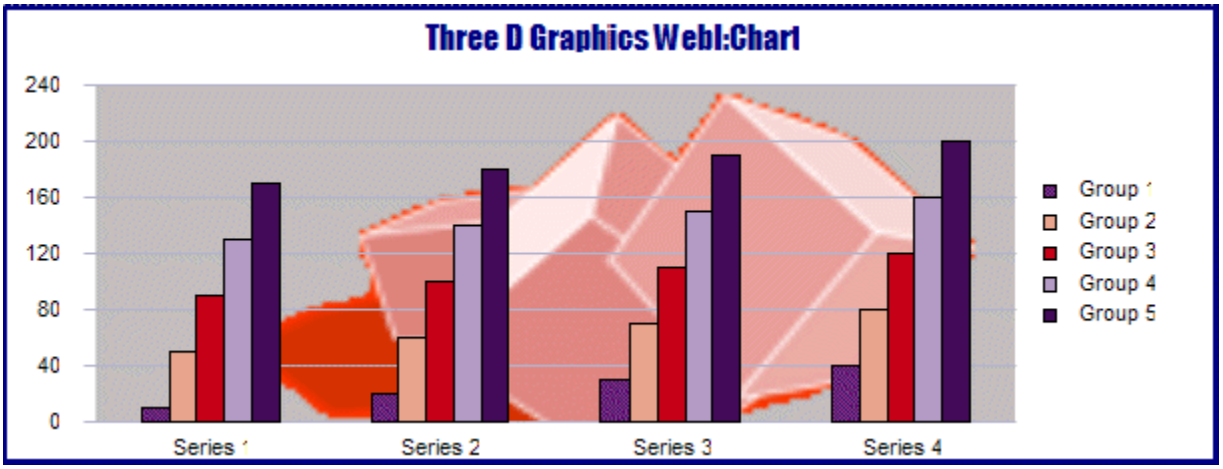
PARAMETERS:

name; Picture name that can be used by other macros to assign the picture to a chart object (e.g., @GCOLOR 0 MyPicture).

file; image file path (absolute or relative to working directory)

EXAMPLE:

```
@PICTF pic1 c:\source\website_bezier\tdg\images\boxes_financial.gif
@GCOLOR 0 pic1
```



@SHADOW (Drop Shadow)

This macro applies a drop shadow effect to an object in the chart

SYNTAX:

```
@SHADOW nObject nXOffset nYOffset
```

PARAMETERS:

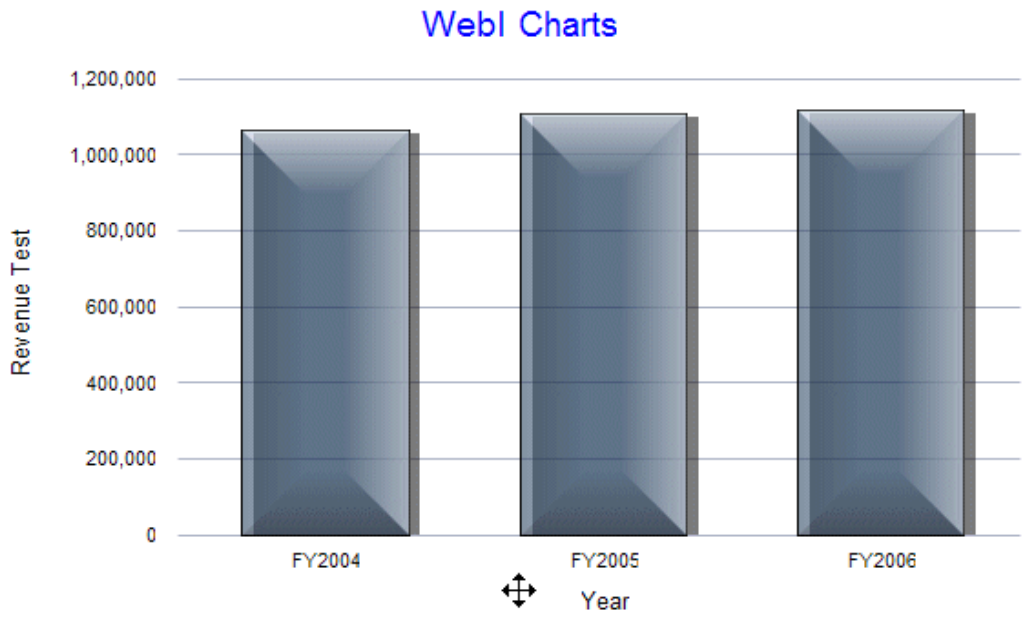
nObject; 0...12 selects one of the following objects:

nObject	*Chart Object*
0	Chart Frame
1	Legend Frame
2	Title
5	Y1 Axis Title
6	Y2 Axis Title
7	X Axis Title
8	Y1 Axis Labels
9	Y2 Axis Labels
10	X Axis Labels
11	Series Labels on Legend
12	Data Markers

nXOffset/*nYOffset;* -1000...1000. If *nXOffset* and/or *nYoffset* are set to 0 (the default), it means there is no shadow.

EXAMPLE:

```
@SHADOW 12 400 -200
```



PERSISTENT:

YES

Section 9: Macros for Lines

These macros can be used to create and format user-defined lines and trend lines:

- @CX; X-Axis Line with Color
- @CXY; Line between two X/Y Coordinates with Color
- @CY; Y-Axis Line with Color
- @CY2; Y-Axis Line with Color, Width, & Line Style
- @INIT_USERLINES; Initialize all user-defined lines
- @MEAN; Enable/disable a mean average line across a specified series
- @TRENDLINE; Trend Line across a series
- @TRENDLINE2; Trend Line across a series with thickness & style
- @X; X-Axis Line at Value
- @XG; X-Axis Line at Group
- @XSZ; X-Axis Line with Label
- @XSZL; X-Axis Line with Label on Left
- @XSZN; X-Axis Line with Label & Value
- @XSZNL; X-Axis Line with Label & Value on Left
- @XY; Line between two X/Y Coordinates
- @XY_DP2; Line between two Data Points (Scatter Charts)
- @Y; Y1-Axis Line
- @YSZ; Y1-Axis Line with Label
- @YSZL; Y1-Axis Line with Label on Left
- @YSZN; Y1-Axis Line with Label & Value
- @YSZN2; Y1-Axis Line with Label & Value above line
- @YSZNL; Y1-Axis Line with Label & Value on Left

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@CX (X-Axis Line with Color)

This macro adds a user-defined line on the X-axis at *fValue*. Use the *nRed*, *nGreen*, and *nBlue* parameters to specify the RGB color of the line. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

```
@CX fValue nRed nGreen nBlue
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the X-axis

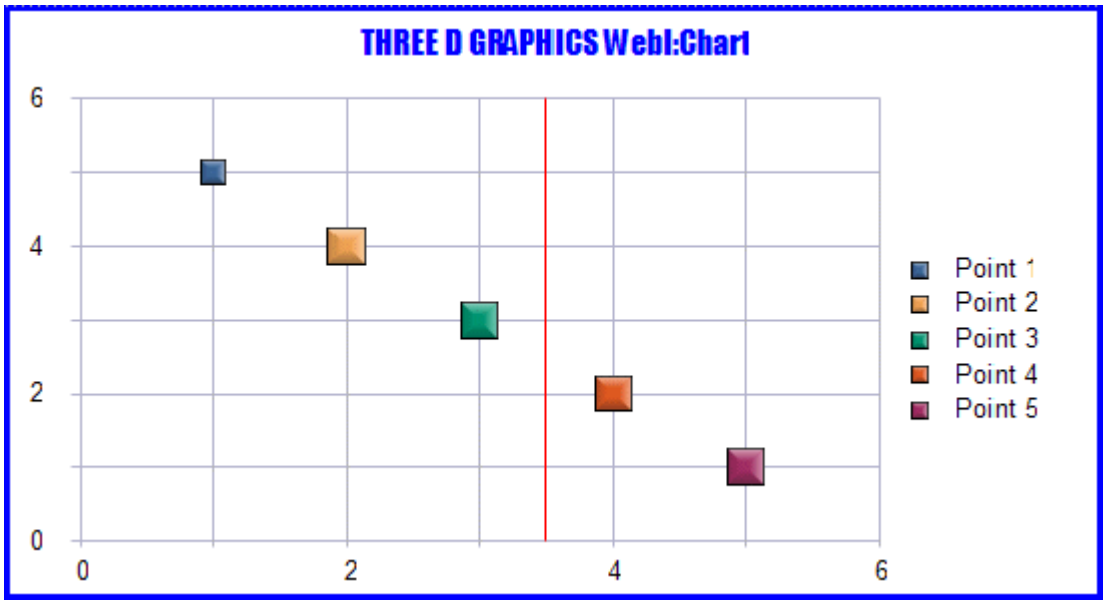
nRed; 0...255 specifies the RED portion of the RGB color

nGreen; 0...255 specifies the GREEN portion of the RGB color

nBlue; 0...255 specifies the BLUE portion of the RGB color

EXAMPLE:

```
@CX 3.5 255 0 0
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@CXY (X/Y Coordinates Line with Color)

This macro adds a user-defined line that starts at location *fXBegin*, *fYBegin* and stops at location *fXEnd*, *fYEnd*. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *fXBegin* and *fXEnd* define the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *fXBegin* and *fXEnd* must be set to a value in the range 0.0 to 1.0 that defines a percentage of the X (or ordinal)-Axis length. Use the *nRed*, *nGreen*, and *nBlue* parameters to specify the RGB color of the line.

SYNTAX:

```
@CXY fXBegin fYBegin fXEnd fYEnd nRed nGreen nBlue
```

PARAMETERS:

fXBegin; Beginning x-coordinate

fYBegin; Beginning y-coordinate

fXEnd; Ending x-coordinate

fYEnd; Ending y-coordinate

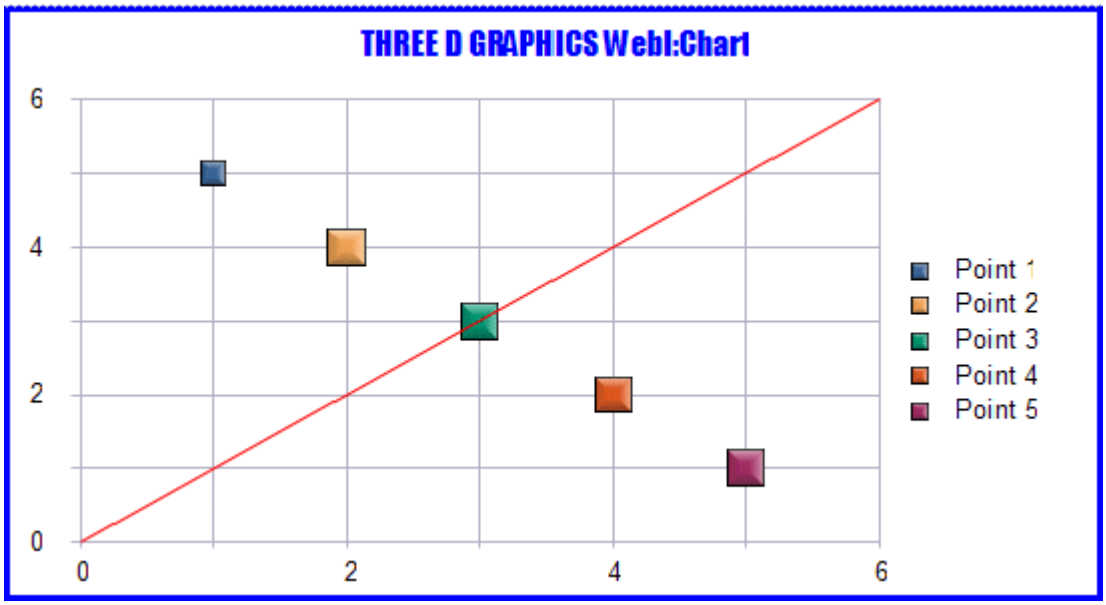
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

EXAMPLE:

```
@CXY 0 0 6 6 255 0 0
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@CY (Y1-Axis Line with Color)

This macro adds a user-defined line on the Y1-axis at value *fYValue*. Use the *nRed*, *nGreen*, and *nBlue* parameters to specify the RGB color of the line. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

```
@CY fYValue nRed nGreen nBlue
```

PARAMETERS:

fYValue; Value at which to add the user-defined line on the Y1-axis

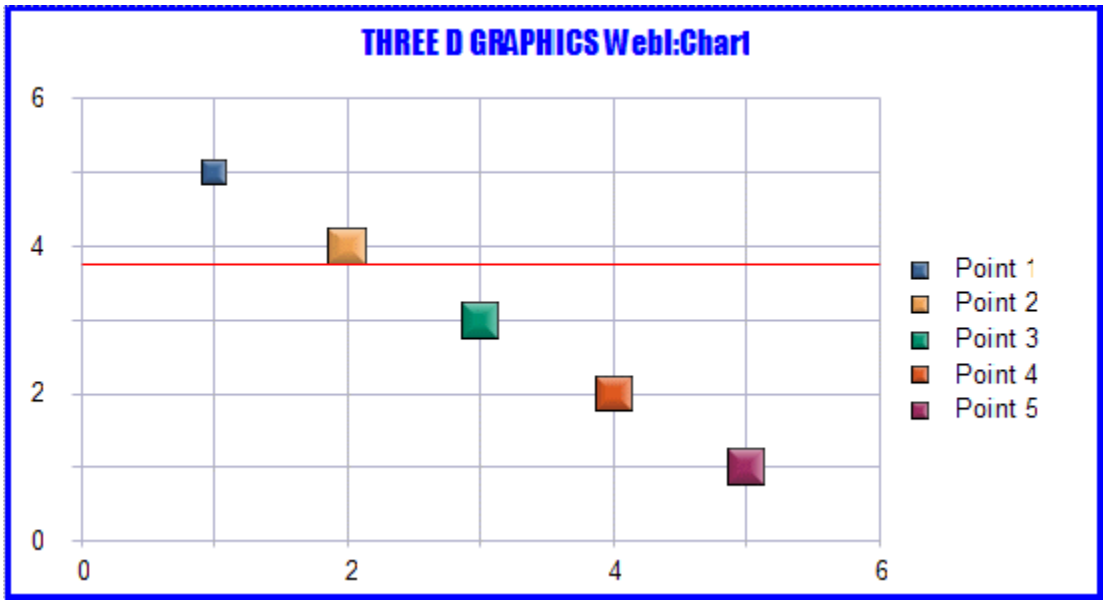
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

EXAMPLE:

```
@CY 3.75 255 0 0
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@CY2 (Y1-Axis Line with Color, Width & Style)

This macro adds a user-defined line on the Y1-axis at value *fYValue*. Use the *nRed*, *nGreen*, and *nBlue* parameters to specify the RGB color of the line. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

```
@CY2 fYValue nRed nGreen nBlue nWidth nStyle
```

PARAMETERS:

fYValue; Value at which to add the user-defined line on the Y1-axis

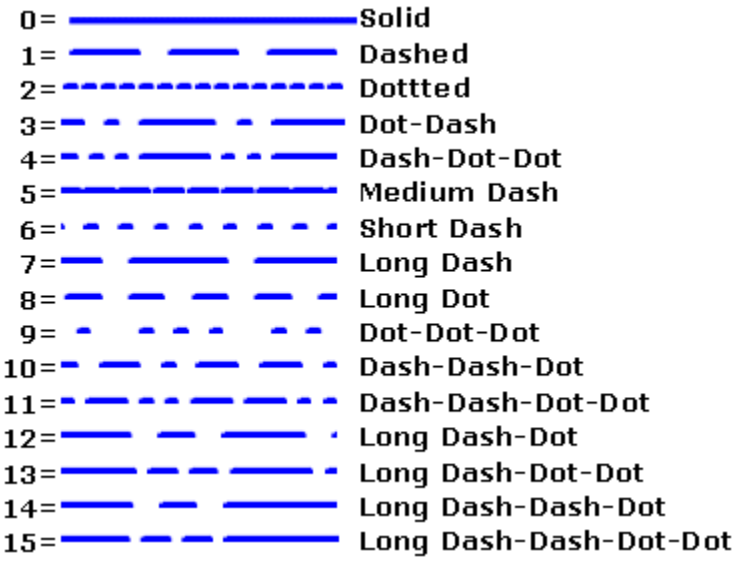
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

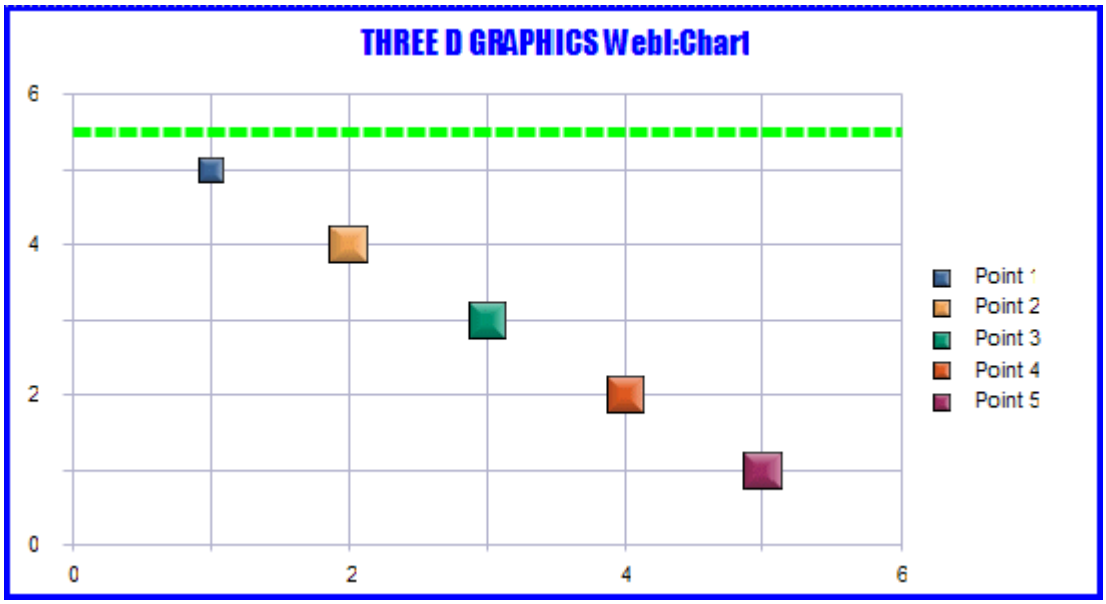
nWidth; 0...1000 selects the thickness of the line.

nLineStyle; 0...15 selects one of the following line styles.

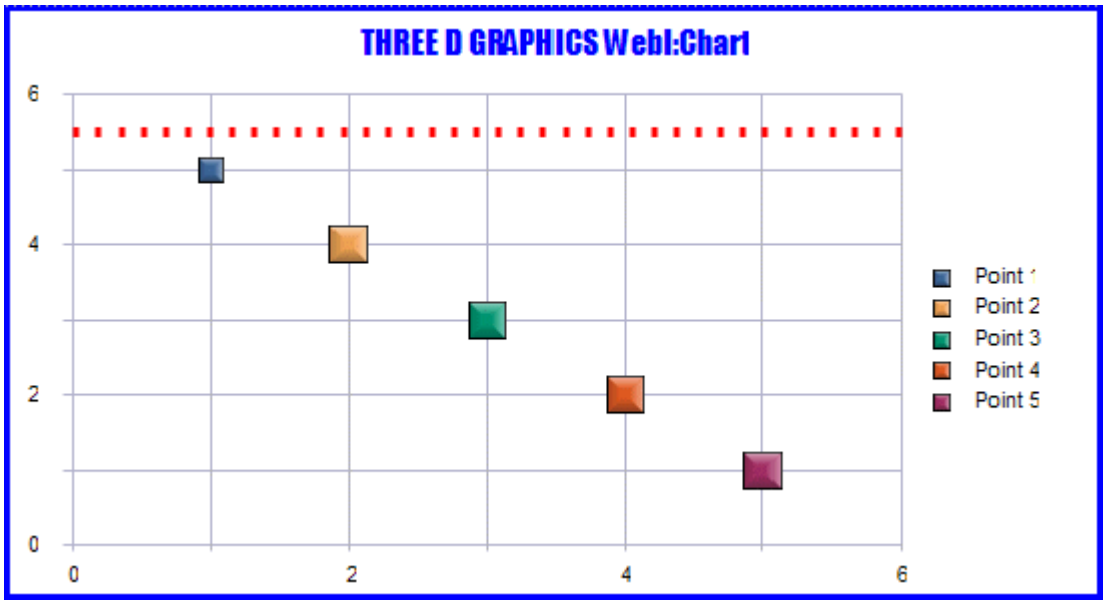


EXAMPLE:

```
@CY2 5.5 0 255 0 500 5
```




```
@CY2 5.5 255 0 0 600 6
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@INIT_USERLINES (Initialize User-Defined Lines)

This macro initializes all user-defined lines to the following parameters: width=1 pixel, pattern=solid, color=BLACK. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

SYNTAX:

```
@INIT_USERLINES
```

PARAMETERS:

None

PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines.

@MEAN (Mean Line)

This macro enables/disables a mean average line across a specified series.

SYNTAX:

```
@MEAN nSeries bShow
```

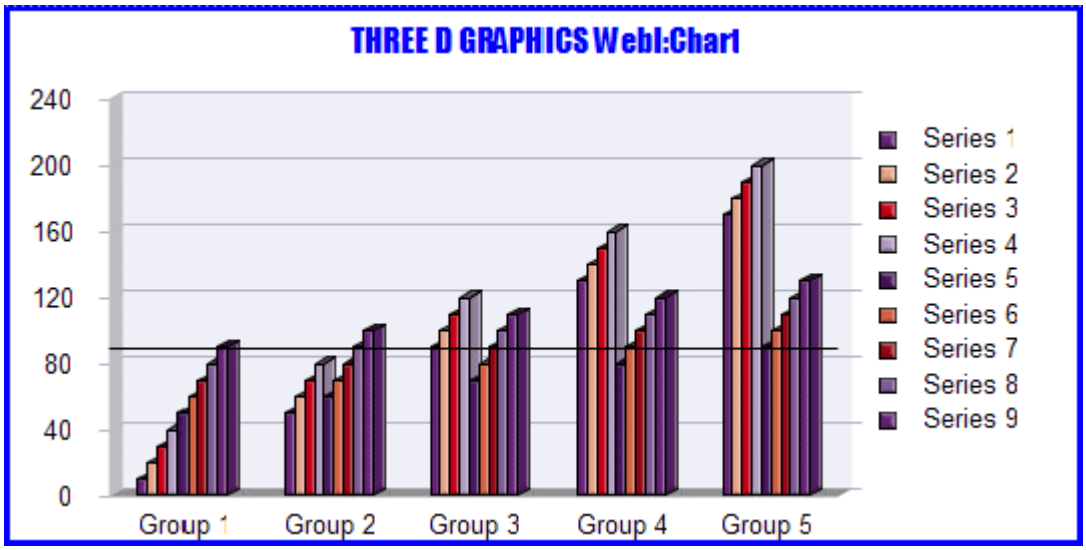
PARAMETERS:

nSeries; 0...number of series in chart (0=Series 1).

bShow; 1 = Draw mean curve line for *nSeries*, 0 = Do not draw mean curve line for *nSeries*.

SYNTAX:

```
@MEAN 0 1
```



PERSISTENT:

YES

@TRENDLINE (Trend Line)

This macro draws a trend line across a specified series.

SYNTAX:

```
@TRENDLINE nSeries nValue
```

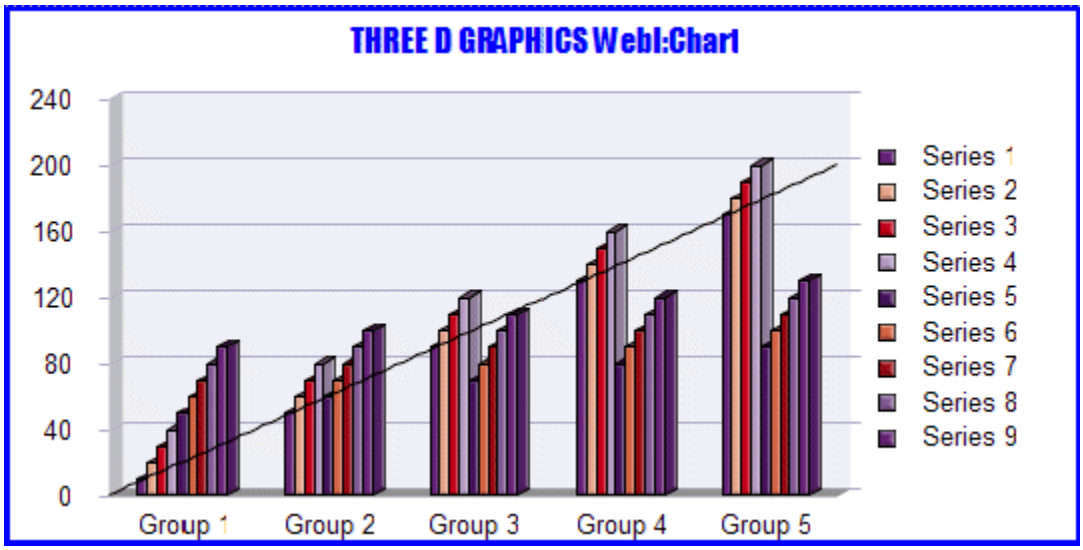
PARAMETERS:

nSeries; 0...511 Series (0 = Series 1) to apply the trend line.

nValue; Bitwise flags (0...127) to activate one or more of the following trend lines:
1 = MEAN
2 = Standard Deviation
4 = Linear Regression
8 = Natural Log Regression
16 = Polynomial Regression
32 = Exponential Regression
64 = Log Regression

EXAMPLE:

```
@TRENDLINE 0 4
```



PERSISTENT:

YES

@TRENDLINE2 (Trend Line with Thickness/Style)

This macro draws a trend line across a specified series with thickness and style.

SYNTAX:

```
@TRENDLINE nSeries nValue nWidth nStyle
```

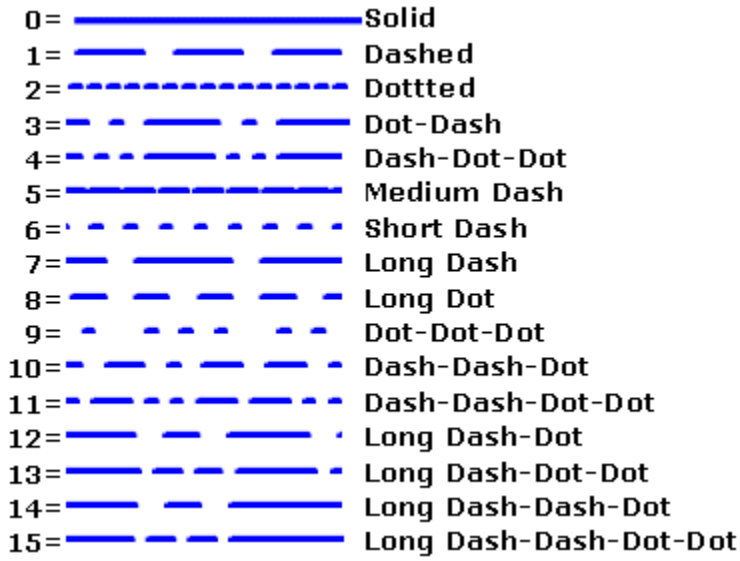
PARAMETERS:

nSeries; 0...511 Series (0 = Series 1) to apply the trend line.

nValue; Bitwise flags (0...127) to activate one or more of the following trend lines:
1 = MEAN
2 = Standard Deviation
4 = Linear Regression
8 = Natural Log Regression
16 = Polynomial Regression
32 = Exponential Regression
64 = Log Regression

nWidth; 0...1000 selects the thickness of the line.

nLineStyle; 0...15 selects one of the following line styles.



PERSISTENT:

YES

@X (X-Axis Line at Value)

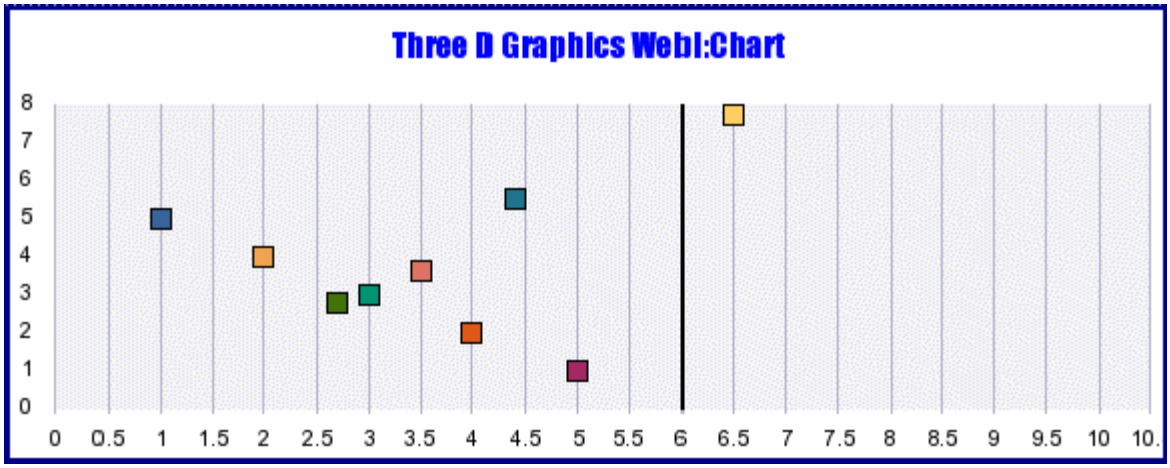
This macro adds a user-defined line on the X-Axis. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *fXValue* defines the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *fXValue* must be set to a value in the range 0.0 to 1.0 that defines a percentage of the X (or ordinal)-Axis length. For example @X .5 will draw a line that is 50% of the distance between the left and right sides of the chart frame. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

```
@X 6
```

PARAMETERS:

fXValue; Value at which to add the user-defined line on the X-axis



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XG (X-Axis Line at Group)

This macro is like the @X macro except it draws a line at a specified group. It allows you place the line more accurately (using *groupID*) on Bar/line/area/Box Plots. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

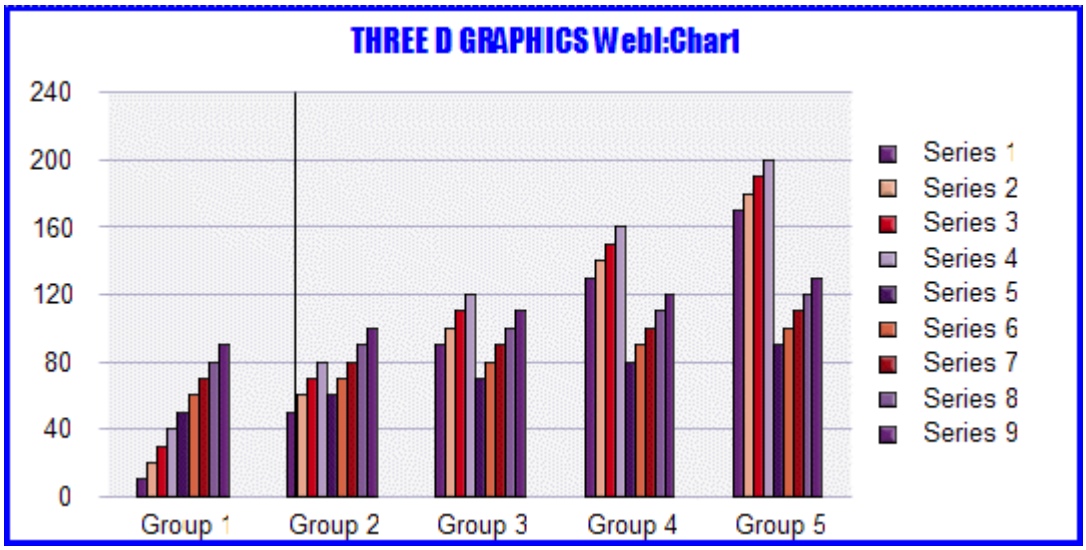
```
@XG groupID
```

PARAMETERS:

groupID; Group at which to add the user-defined line on the X-axis

EXAMPLE:

```
@XG 1
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XSZ (X-Axis Line with Label)

This macro adds a user-defined line on the X-axis at value *fValue* with the label *sZLabel*. For vertical orientation, the line is drawn vertically with the label on the top side of the chart. For horizontal orientation, the line is drawn horizontally with the label on the right side of the chart.

SYNTAX:

```
@XSZ fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the X-axis

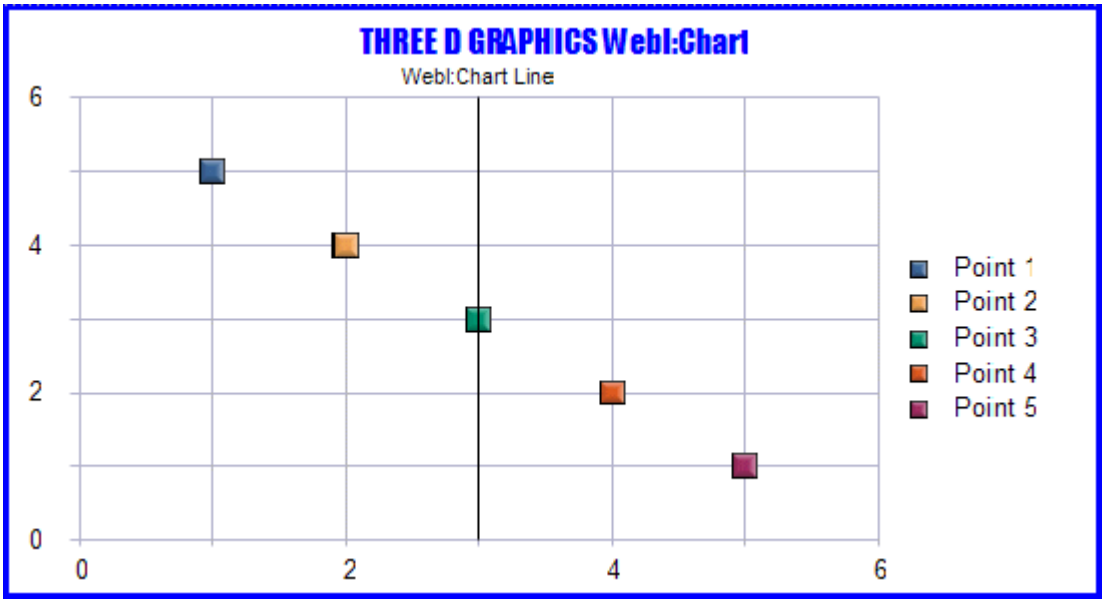
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to the line at *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@XSZ 3 9 9 WebI:Chart Line
```



PERSISTENT:

NO

ALSO SEE:

@XSZL, @XSZN, @XSZNL

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XSZL (X-Axis Line with Label on Left/Lower)

This macro adds a user-defined line on the X-axis at value *fValue* with the label *sZLabel*. For vertical orientation, the line is drawn vertically with the label on the bottom side of the chart. For horizontal orientation, the line is drawn horizontally with the label on the left side of the chart.

SYNTAX:

```
@XSZL fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the X-axis

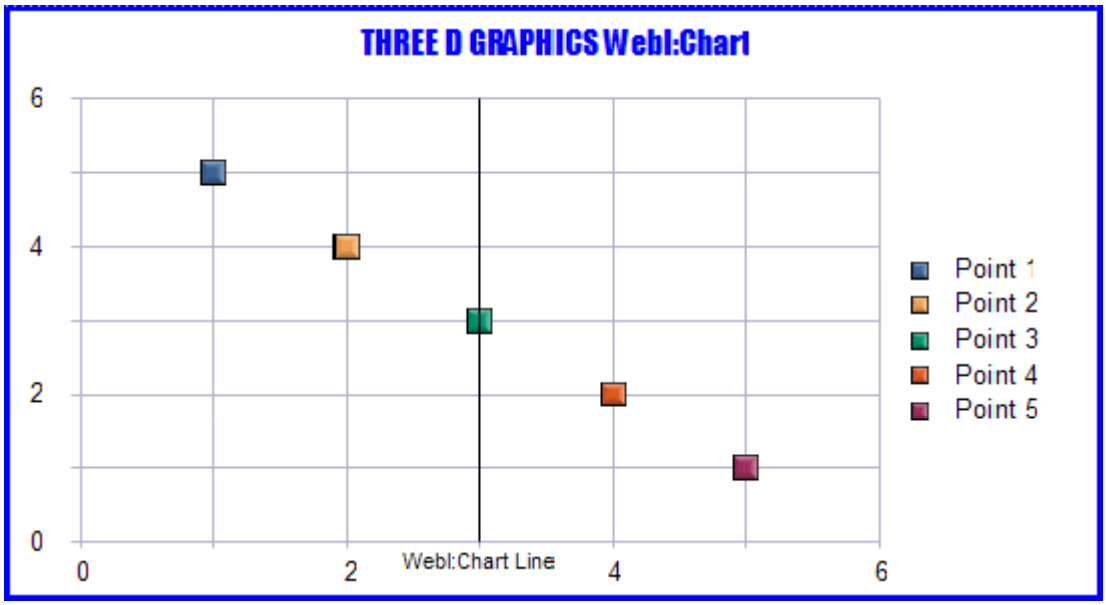
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to line. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@XSZL 3 9 9 WebI:Chart Line
```



PERSISTENT:

NO

ALSO SEE:

@XSZ, @XSZN, @XSZNL

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XSZN (X-Axis Line with Label & Value)

This macro adds a user-defined line on the X-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*. For vertical orientation, the line is drawn vertically with the label and value on the top side of the chart. For horizontal orientation, the line is drawn horizontally with the label and value on the right side of the chart.

SYNTAX:

```
@XSZN fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the X-axis

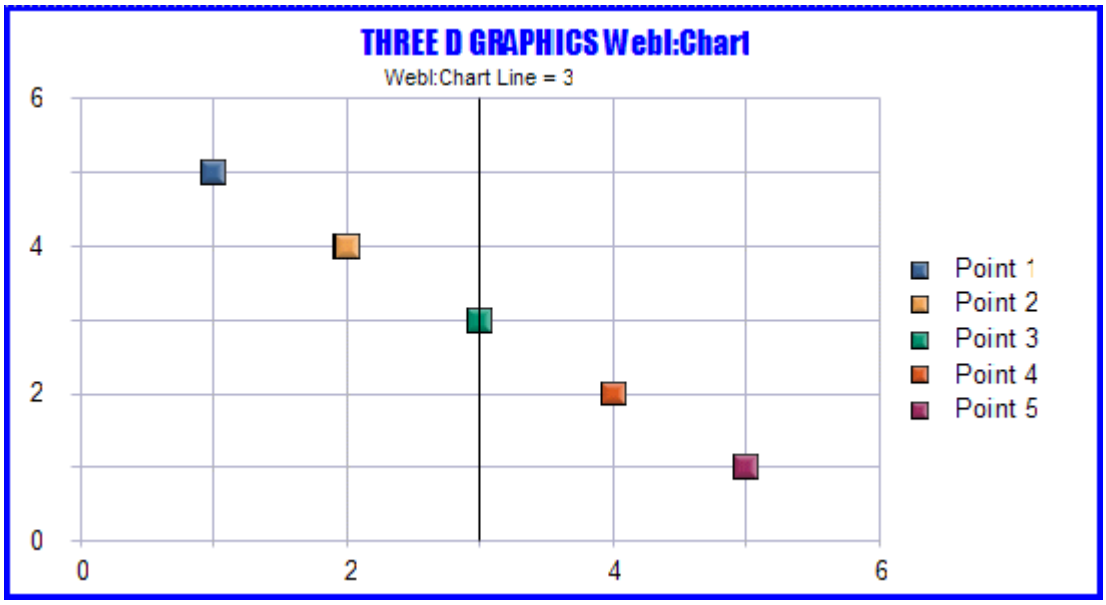
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to line. Add a space to the end of the label if you want a space to appear between the label and *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@XSZN 3 9 9 WebI:Chart Line =
```



PERSISTENT:

NO

ALSO SEE:

@XSZ, @XSZL, @XSZNL

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XSZNL (X-Axis Line with Label & Value on Left/Lower)

This macro adds a user-defined line on the X-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*. For vertical orientation, the line is drawn vertically with the label and value on the bottom side of the chart. For horizontal orientation, the line is drawn horizontally with the label and value on the left side of the chart.

SYNTAX:

```
@XSZNL fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the X-axis

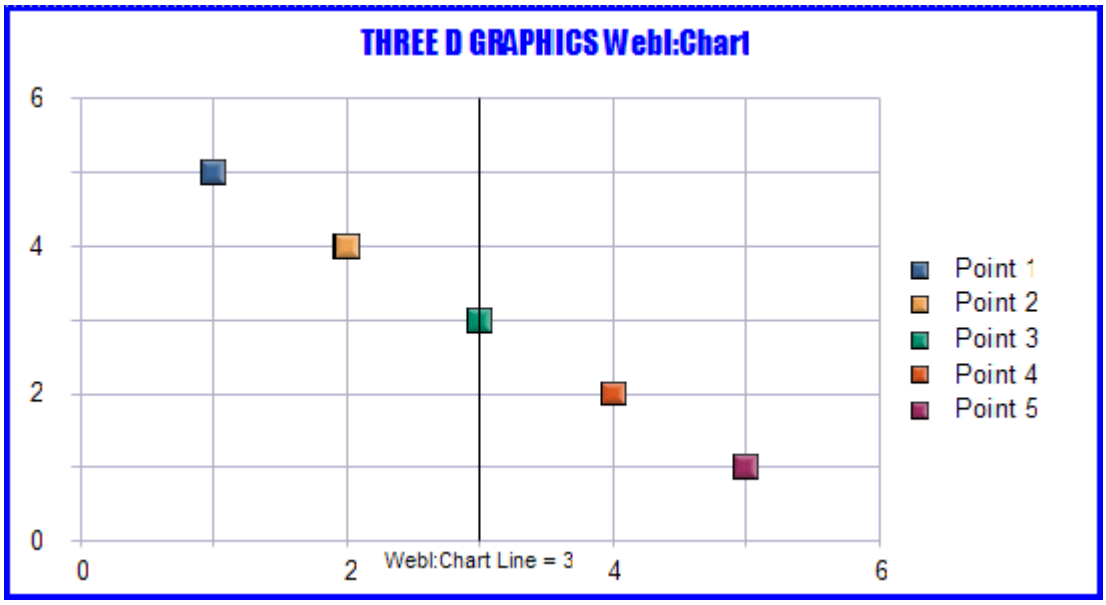
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to the line at *fValue*. Add a space to the end of the label if you want a space to appear between the label and *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@XSZNL 3 9 9 WebI:Chart Line =
```



PERSISTENT:

NO

ALSO SEE:

@XSZ, @XSZL, @XSZN

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XY (X/Y Coordinates Line)

This macro adds a user-defined line that starts at location *fXBegin*, *fYBegin* and stops at location *fXEnd*, *fYEnd*. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *fXBegin* and *fXEnd* define the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *fXBegin* and *fXEnd* must be set to a value in the range 0.0 to 1.0 that defines a percentage of the X (or ordinal)-Axis length.

SYNTAX:

```
@XY fXBegin fYBegin fXEnd fYEnd
```

PARAMETERS:

fXBegin; Beginning X-coordinate

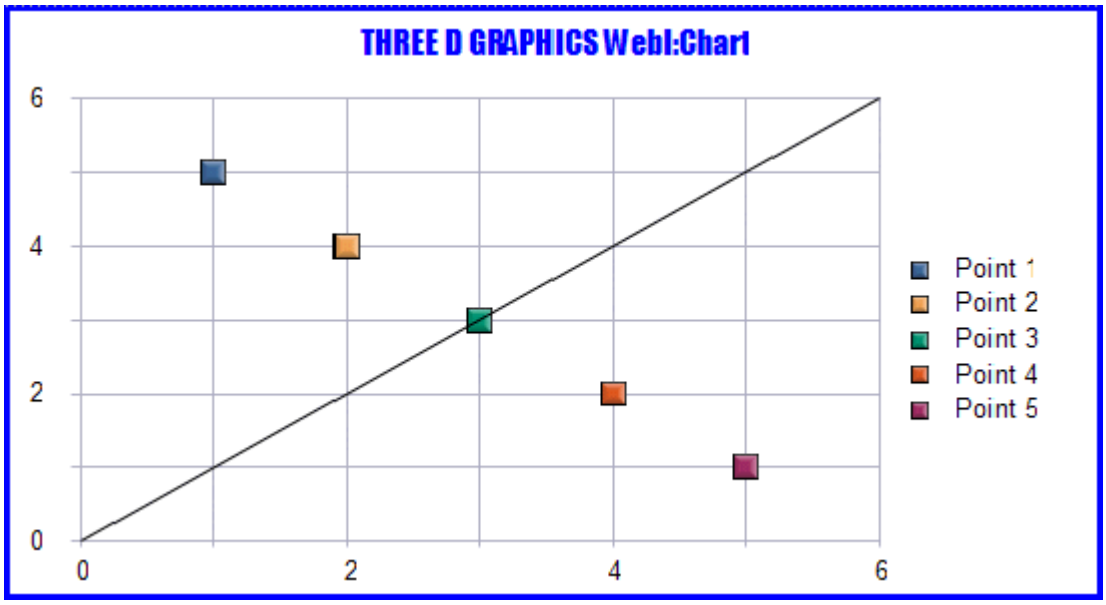
fYBegin; Beginning Y-coordinate

fXEnd; Ending X-coordinate

fYEnd; Ending Y-coordinate

EXAMPLE:

```
@XY 0 0 6 6
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@XY_DP2 (Data Point Line)

On a scatter chart, this macro draws a user-defined line between any two points. The points are specified in terms of series and groups.

SYNTAX:

```
@XY_DP2 nSeriesStart nGroupStart nSeriesStop nGroupStop
```

PARAMETERS:

nSeriesStart; 0...1024 Zero-based series number to start drawing line.

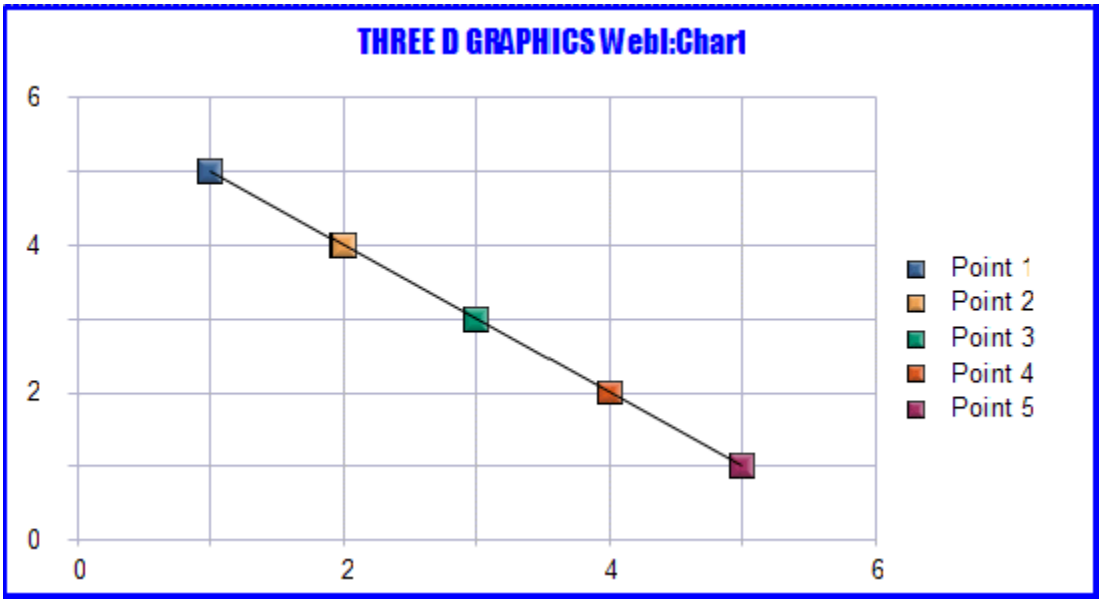
nGroupStart; 0...1024 Zero-based group number to start drawing line.

nSeriesStop; 0...1024 Zero-based series number to stop drawing line.

nGroupStop; 0...1024 Zero-based group number to stop drawing line.

EXAMPLE:

```
@XY_DP2 0 0 0 4
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@Y (Y1-Axis Line)

This macro adds a user-defined line on the Y1-axis at value *fY1Value*. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

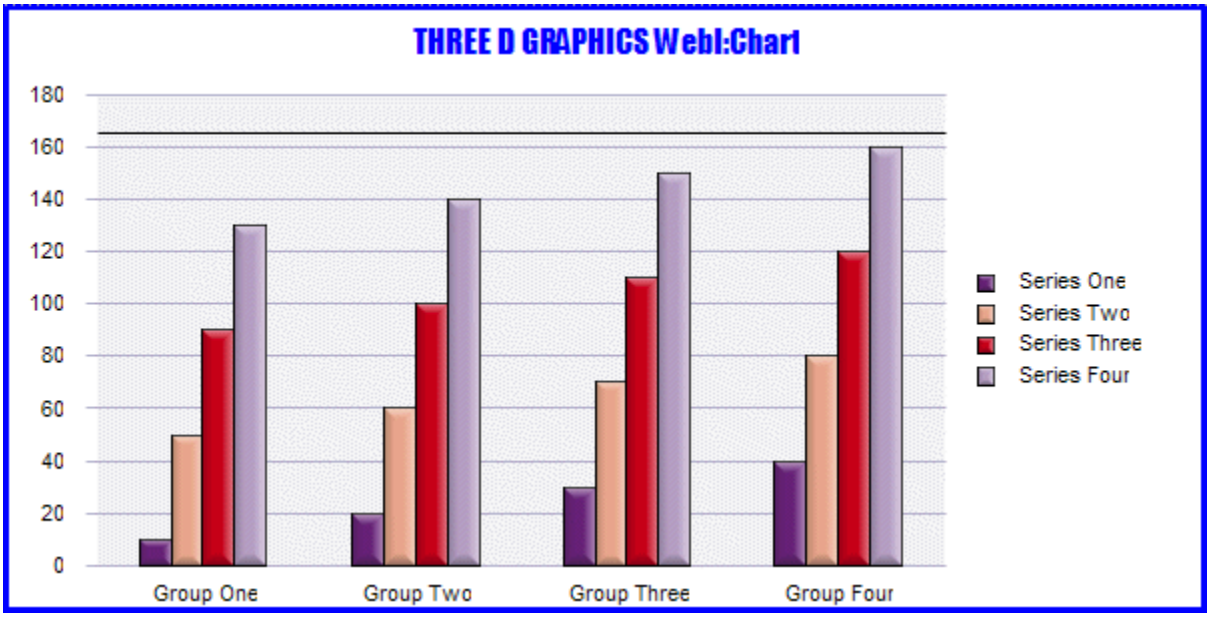
```
@Y fY1Value
```

PARAMETERS:

fY1Value; Value at which to add the user-defined line on the Y1-axis

EXAMPLE:

```
@Y 165
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@YSZ (Y1-Axis Line with Label)

This macro adds a user-defined line on the Y1-axis at value *fValue* with the label *sZLabel*. For vertical orientation, the line is drawn horizontally with the label on the right side of the chart. For horizontal orientation, the line is drawn vertically with the label on the bottom side of the chart.

SYNTAX:

```
@YSZ fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the Y1-axis

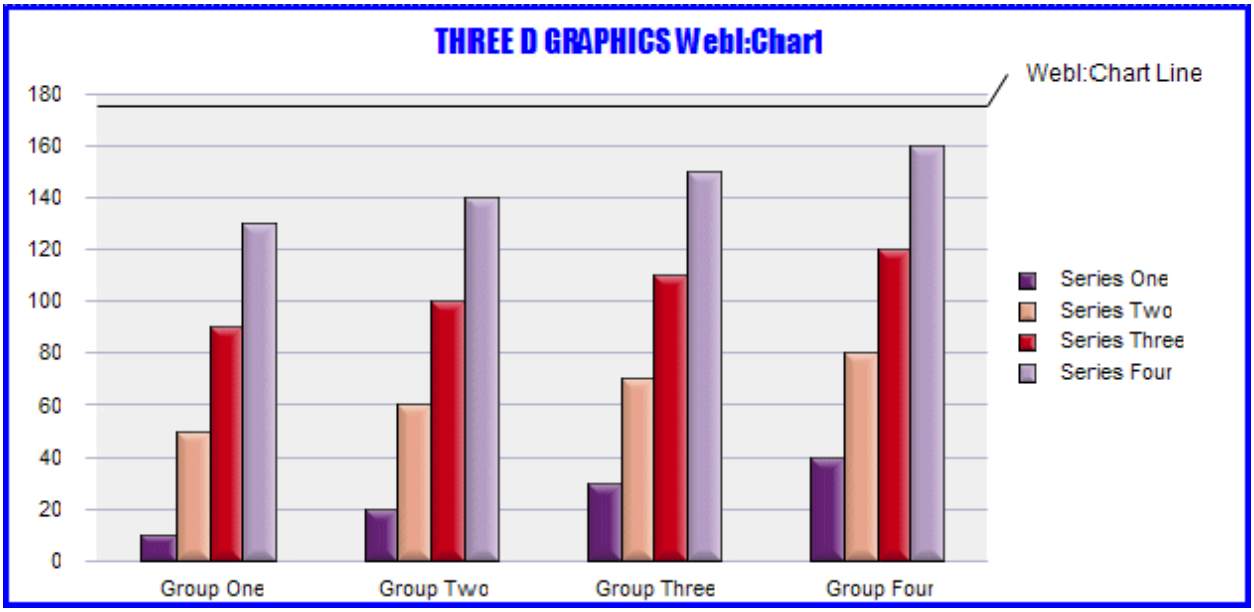
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to the line at *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@YSZ 175 0 0 WebI:Chart Line
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

ALSO SEE:

@YSZL, @YSZN, & @YSZNL

@YSZL (Y1-Axis Line with Label on Left)

This macro adds a user-defined line on the Y1-axis at value *fValue* with the label *sZLabel*. For vertical orientation, the line is drawn horizontally with the label on the left side of the chart. For horizontal orientation, the line is drawn vertically with the label and value on the top side of the chart.

SYNTAX:

```
@YSZL fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the Y1-axis

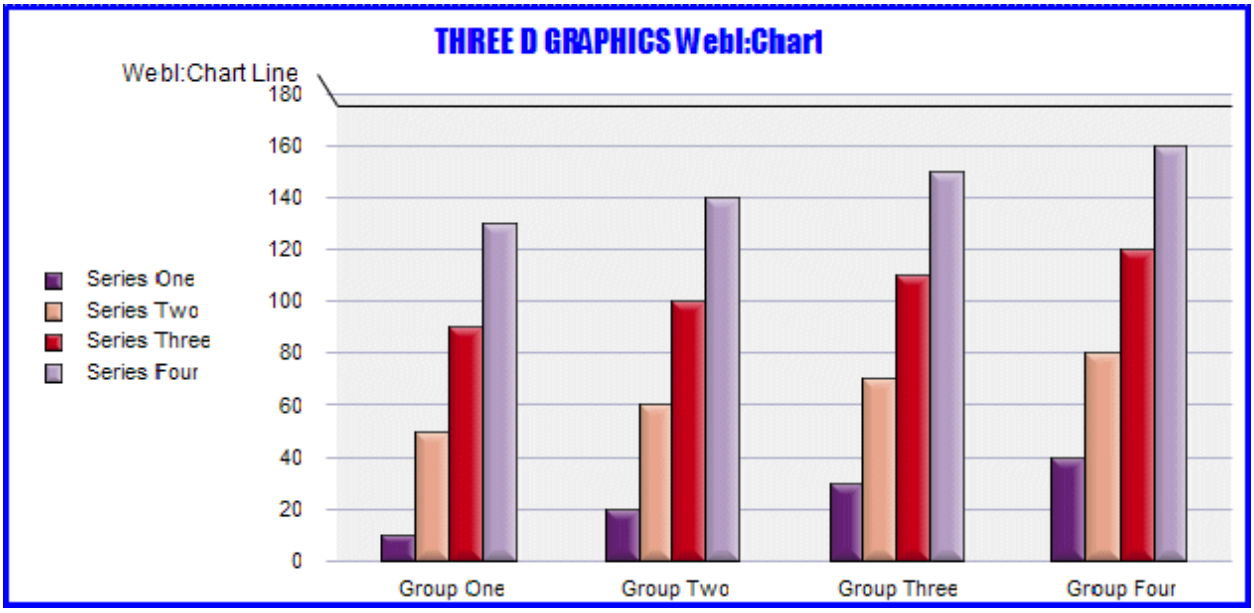
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to the line at *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@YSZL 175 0 0 WebI:Chart Line
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

ALSO SEE:

@YSZ, @YSZN & @YSZNL

@YSZN (Y1-Axis Line with Label & Value)

This macro adds a user-defined line on the Y1-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*. For vertical orientation, the line is drawn horizontally with the label and value on the right side of the chart. For horizontal orientation, the line is drawn vertically with the label and value on the bottom side of the chart.

SYNTAX:

```
@YSZN fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the Y1-axis

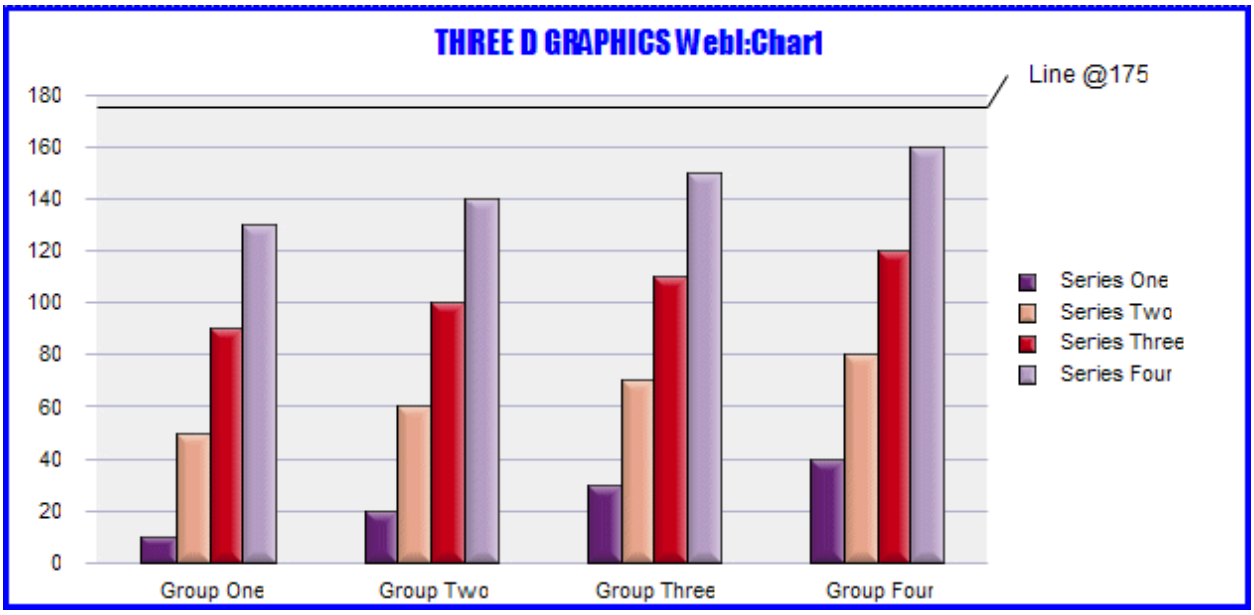
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to the line at *fValue*. Add a space to the end of the label if you want a space to appear between the label and *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@YSZN 175 0 0 Line @
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

ALSO SEE:

@YSZ, @YSZL, & @YSZNL

@YSZN2 (Y1-Axis Line with Label & Value Above Line)

This macro adds a user-defined line on the Y1-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*. For vertical orientation, the line is drawn horizontally with the label and value on the right side of the chart. For horizontal orientation, the line is drawn vertically with the label and value on the bottom side of the chart.

SYNTAX:

```
@YSZN fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the Y1-axis

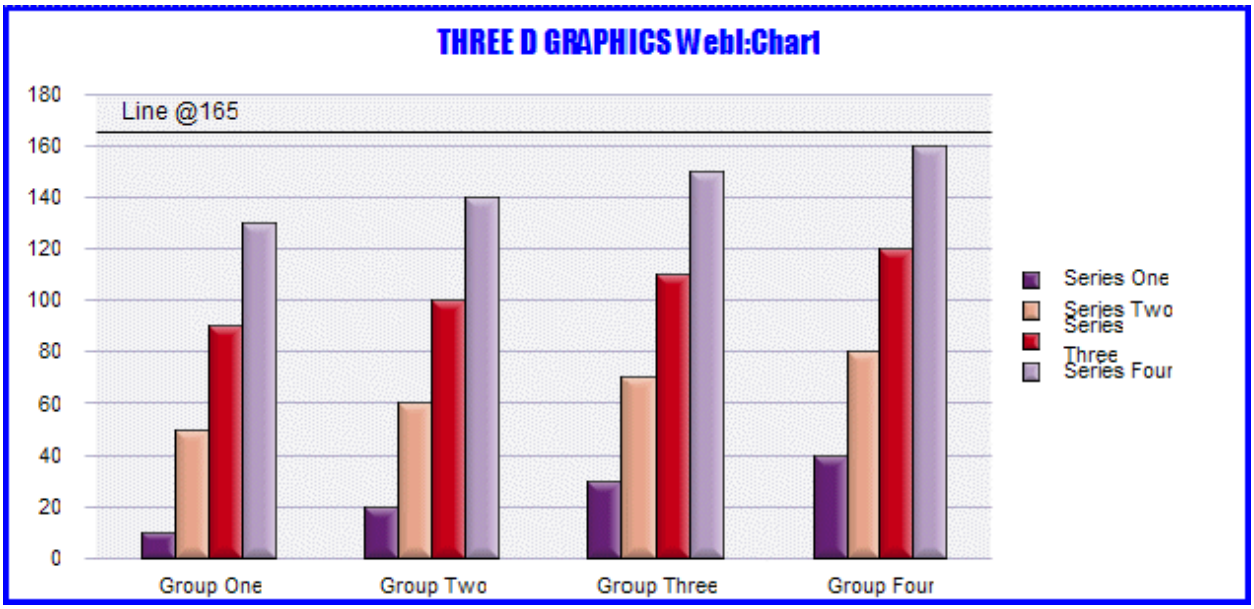
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to the line at *fValue*. Add a space to the end of the label if you want a space to appear between the label and *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@YSZN2 165 0 0 Line @
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

@YSZNL (Y1-Axis Line with Label & Value on Left)

This macro adds a user-defined line on the Y1-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*. For vertical orientation, the line is drawn horizontally with the label and value on the left side of the chart. For horizontal orientation, the line is drawn vertically with the label and value on the top side of the chart.

SYNTAX:

```
@YSZNL fValue nXAdjust nYAdjust sZLabel
```

PARAMETERS:

fValue; Value at which to add the user-defined line on the Y1-axis

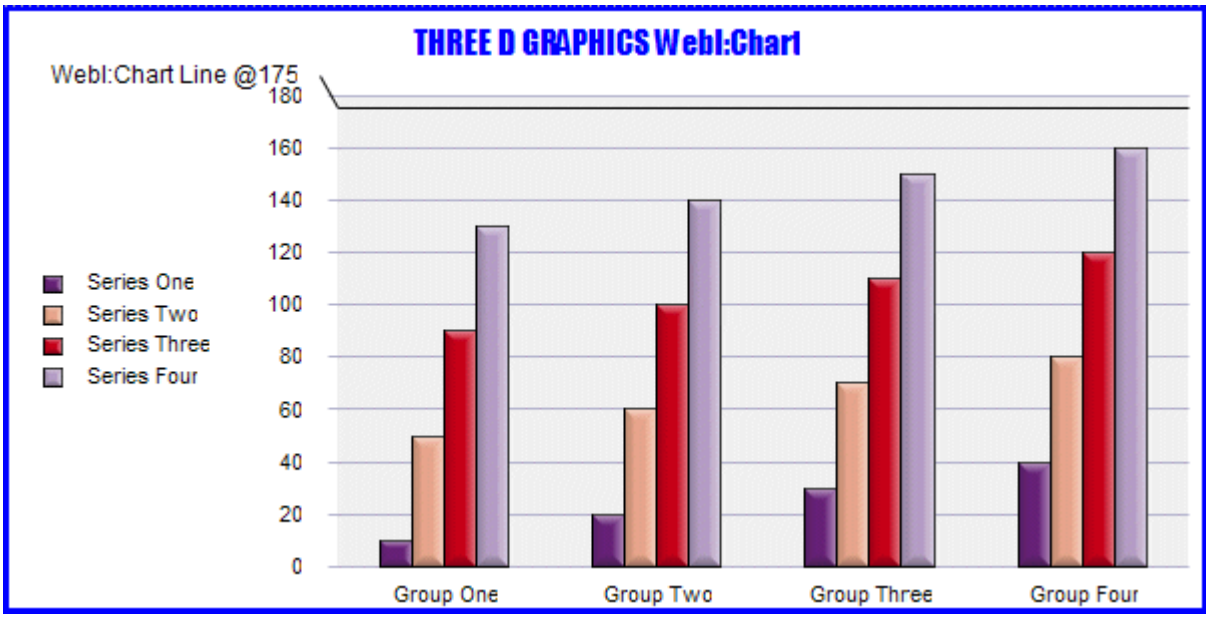
nXAdjust; -16000...16000 adjusts the label closer to or further away from the line.

nYAdjust; -16000...16000 adjusts the label closer to or further away from the line.

sZLabel; Label string to show next to line. Add a space to the end of your label if you want a space to appear between the label and *fValue*. Add a tilde (~) character to the end of the string if you intend to define other macros in the same title field.

EXAMPLE:

```
@YSZNL 175 0 0 WebI:Chart Line @
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @CY2, @X, @XG, @XSZ, @XSZN, @XSZL, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZN2, @YSZL, and @YSZNL.

ALSO SEE:

@YSZ, @YSZL, & @YSZN

Section 10: User-Defined Areas

These macros can be used to draw user-defined circles, rectangles, and markers. Circles and rectangles can be outlined, color filled, or pattern filled and can be drawn in front of or behind the chart area:

- @UF; Same as @USER_FILL
- @USER_CIRCLE; Draw a User-Defined Outlined Circle
- @USER_CIRCLE_ABOVE; Draw a User-Defined Outlined Circle above the Chart Area
- @USER_FILL; Draw a User-Defined Color-Filled Rectangle
- @USER_FILL_CIRCLE; Draw a User-Defined Color-Filled Circle
- @USER_FILL_CIRCLE_ABOVE; Draw a User-Defined Color-Filled Circle above the Chart Area
- @USER_FILL_CIRCLE2; Draw a User-Defined Pattern-Filled Circle
- @USER_FILL_CIRCLE2_ABOVE; Draw a User-Defined Pattern-Filled Circle Above the Chart Area
- @USER_FILL2; Draw a User-Defined Pattern-Filled Rectangle
- @USER_MARKER; Draw a User-Defined Marker at X/Y coordinates
- @USER_RECT; Draw a User-Defined Outlined Rectangle
- @UW; Draw a vertical band
- @WC; Color @UW vertical band

@USER_CIRCLE (Outlined Circle)

On 2D charts, this macro fills a portion of the chart frame with a circle. Set *fStartX/ fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_CIRCLE fStartX fStopX fStartY fStopY nRed nGreen nBlue
nLineStyle nThickness szPhrase
```

PARAMETERS:

fStartX; 0.0...1.1 X-Axis start location

fStopX; 0.0...1.1 X-Axis stop location

fStartY; 0.0...1.1 Y-Axis start location

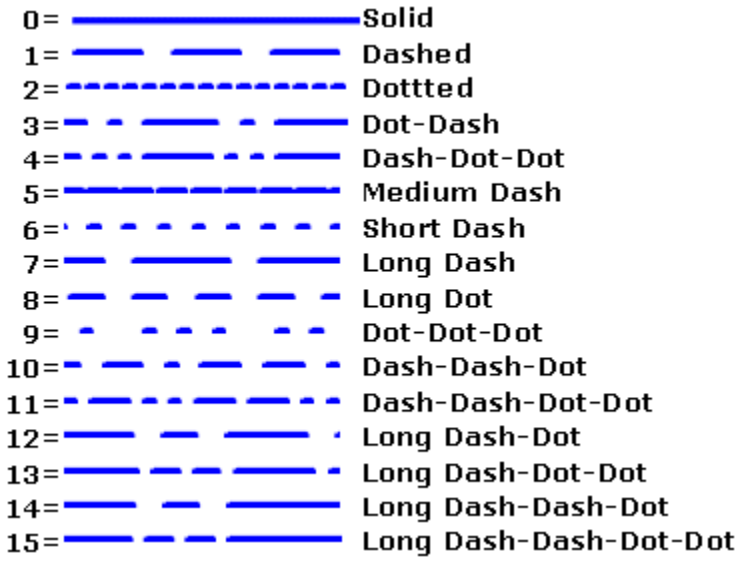
fStopY; 0.0...1.1 Y-Axis stop location

nRed; 0...255 defines the RED portion of the RGB color

nGreen; 0...255 defines the GREEN portion of the RGB color

nBlue; 0...255 defines the BLUE portion of the RGB color

nLineStyle; 0...15 selects the outline style

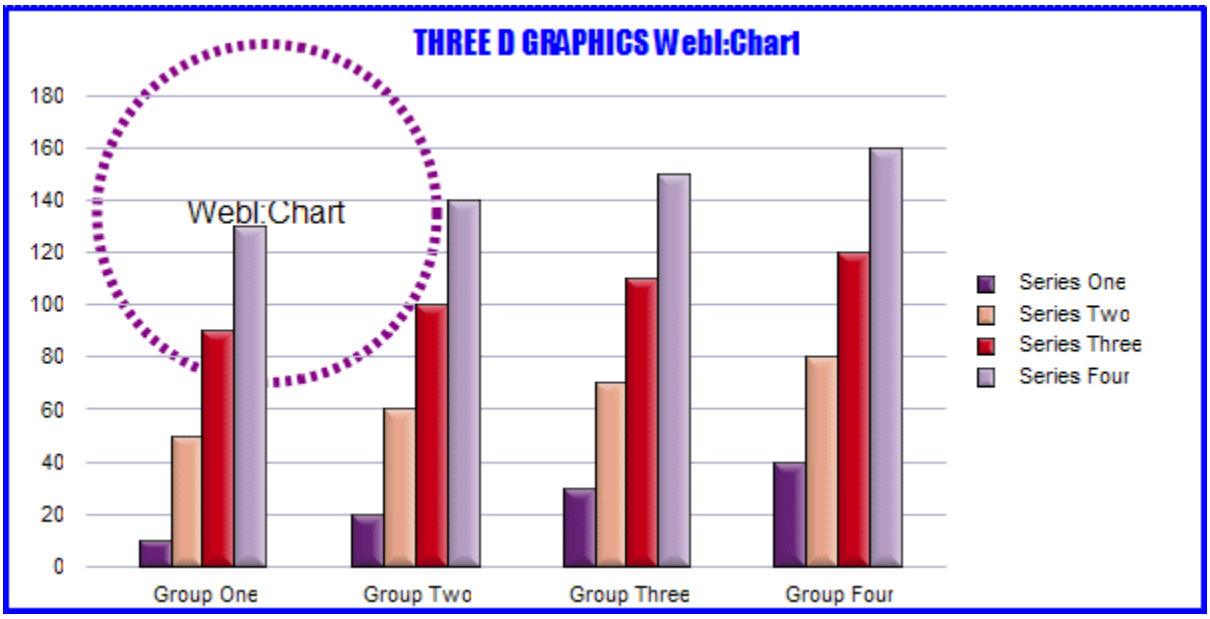


nThickness; 0...1000 selects the thickness of the outline.

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_CIRCLE 0 0.0 0.4 0.4 1.1 128 0 128 2 500 WebI:Chart
```



PERSISTENT:

NO

NOTES:

- WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW.
- @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_CIRCLE_ABOVE (Outlined Circle above Chart Area)

On 2D charts, this macro fills a portion of the chart frame with a circle. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_CIRCLE_ABOVE fStartX fStopX fStartY fStopY nRed nGreen nBlue
nLineStyle nThickness szPhrase
```

PARAMETERS:

fStartX; 0.0...1.1 X-Axis start location

fStopX; 0.0...1.1 X-Axis stop location

fStartY; 0.0...1.1 Y-Axis start location

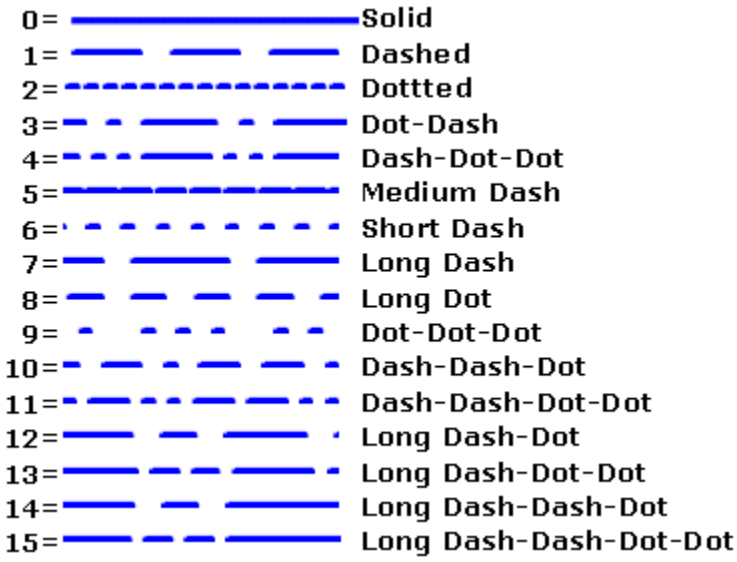
fStopY; 0.0...1.1 Y-Axis stop location

nRed; 0...255 defines the RED portion of the RGB color

nGreen; 0...255 defines the GREEN portion of the RGB color

nBlue; 0...255 defines the BLUE portion of the RGB color

nLineStyle; 0...15 selects the outline style

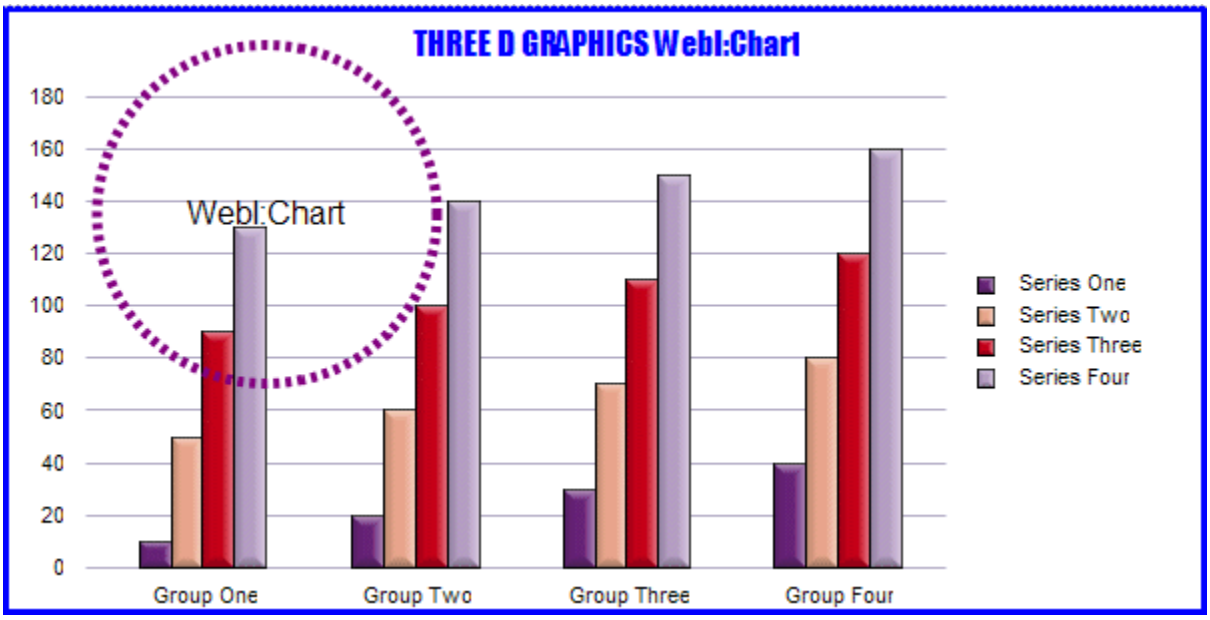


nThickness; 0...1000 selects the thickness of the outline.

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_CIRCLE_ABOVE 0 0.0 0.4 0.4 1.1 128 0 128 2 500 WebI:Chart
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW. @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@UF/@USER_FILL (Color-Filled Rectangle)

On 2D charts, these macros fill a portion of the chart frame with a specified color and optional phrase. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@UF fStartX fStopX fStartY fStopY nRed nGreen nBlue szPhrase
```

or

```
@USER_FILL fStartX fStopX fStartY fStopY nRed nGreen nBlue szPhrase
```

PARAMETERS:

fStartX/*fStopX*; 0.0...1.1 X-Axis start/stop locations

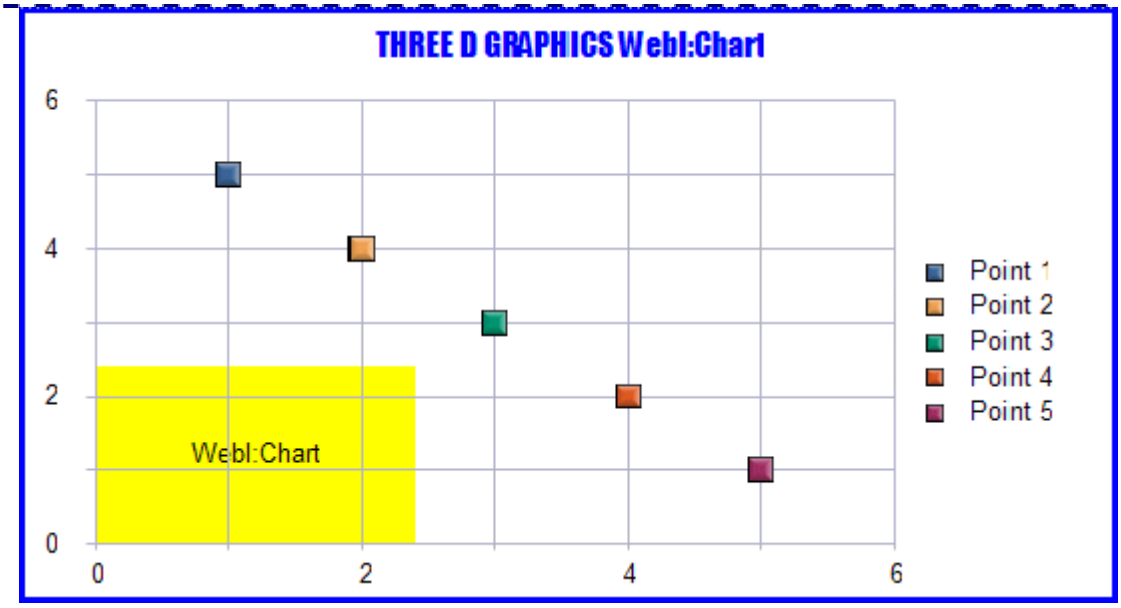
fStartY/*fStopY*; 0.0...1.1 Y-Axis start/stop locations

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_FILL 0 0.4 0 0.4 128 128 0 WebI:Chart
```



PERSISTENT:

NO

NOTES:

- WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW.
- @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_FILL2 (Pattern-Filled Rectangle)

On 2D charts, this macro fills a portion of the chart frame with a specified color, pattern, and optional phrase. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_FILL2 fStartX fStopX fStartY fStopY nRed nGreen nBlue nPattern
szPhrase
```

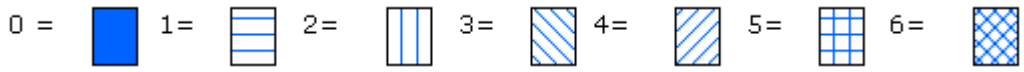
PARAMETERS:

fStartX/fStopX; 0.0...1.1 X-Axis start/stop location

fStartY/fStopY; 0.0...1.1 Y-Axis start/stop location

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

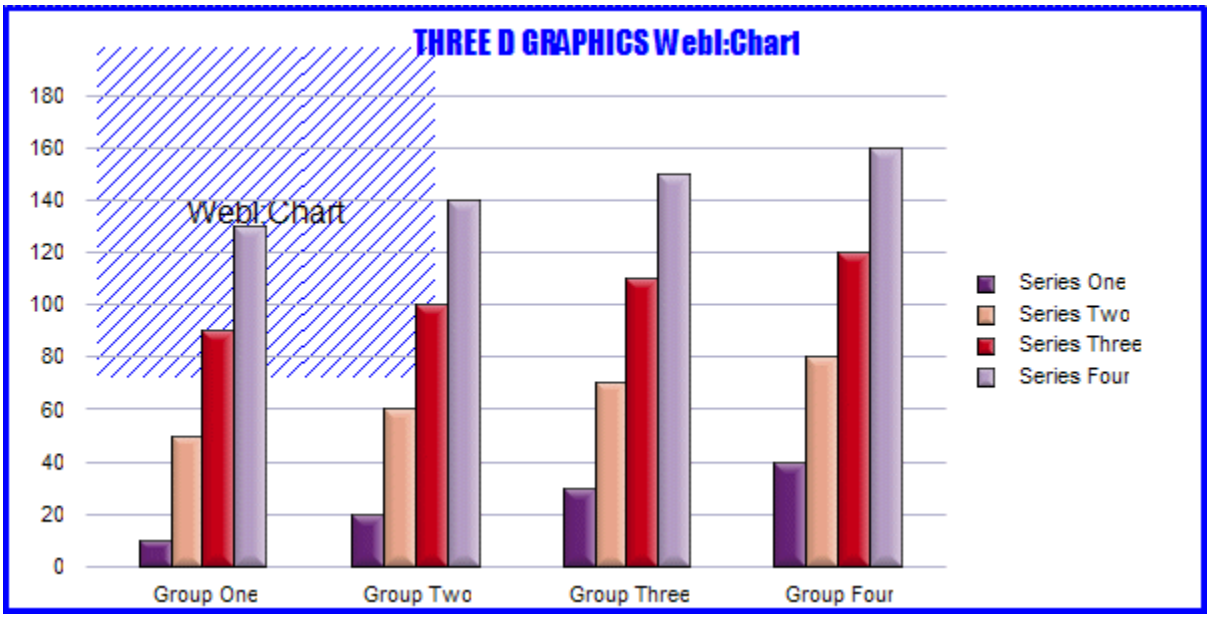
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_FILL2 0 0.0 0.4 0.4 1.1 0 0 255 -4 WebI:Chart
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW. @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_FILL_CIRCLE (Color-Filled Circle)

On 2D charts, this macro fills a portion of the chart frame with a circle that is filled with a specified color and optional phrase. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_FILL_CIRCLE fStartX fStopX fStartY fStopY nRed nGreen nBlue
szPhrase
```

PARAMETERS:

fStartX; 0.0...1.1 X-Axis start location

fStopX; 0.0...1.1 X-Axis stop location

fStartY; 0.0...1.1 Y-Axis start location

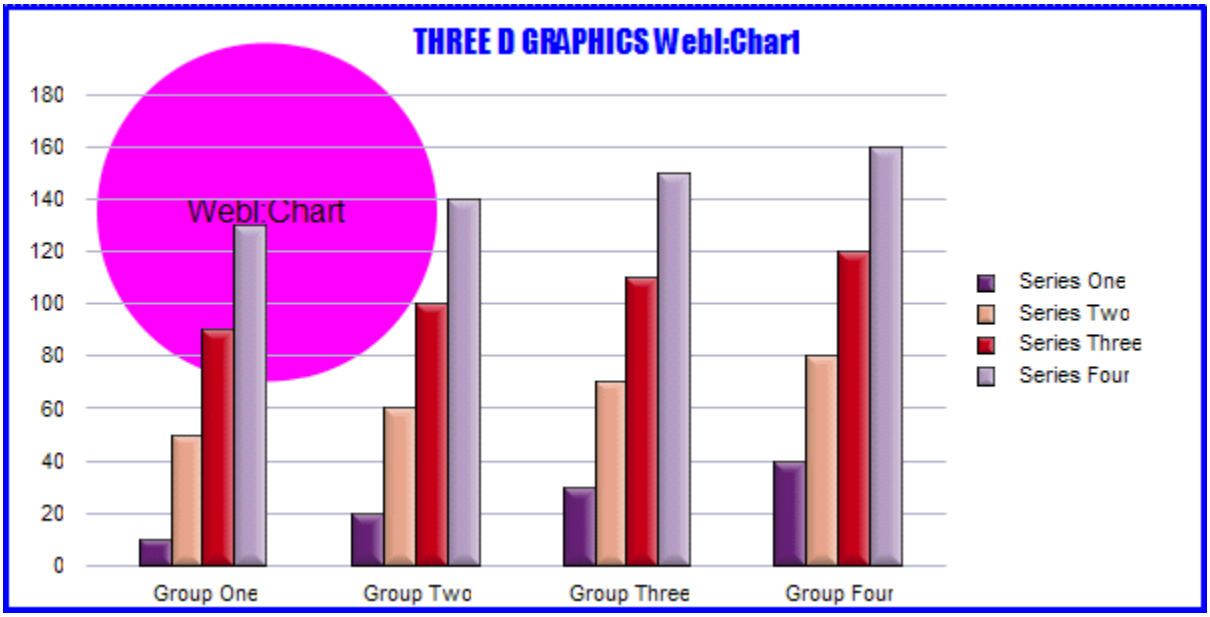
fStopY; 0.0...1.1 Y-Axis stop location

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_FILL_CIRCLE 0 0.0 0.4 0.4 1.1 255 0 255 WebI:Chart
```



PERSISTENT:

NO

NOTES:

- WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW.
- @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_FILL_CIRCLE_ABOVE (Color-Filled Circle above Chart Area)

On 2D charts, this macro fills a portion of the chart frame with a circle that is filled with a specified color and optional phrase. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_FILL_CIRCLE_ABOVE fStartX fStopX fStartY fStopY nRed nGreen
nBlue szPhrase
```

PARAMETERS:

fStartX/fStopX; 0.0...1.1 X-Axis start/stop locations

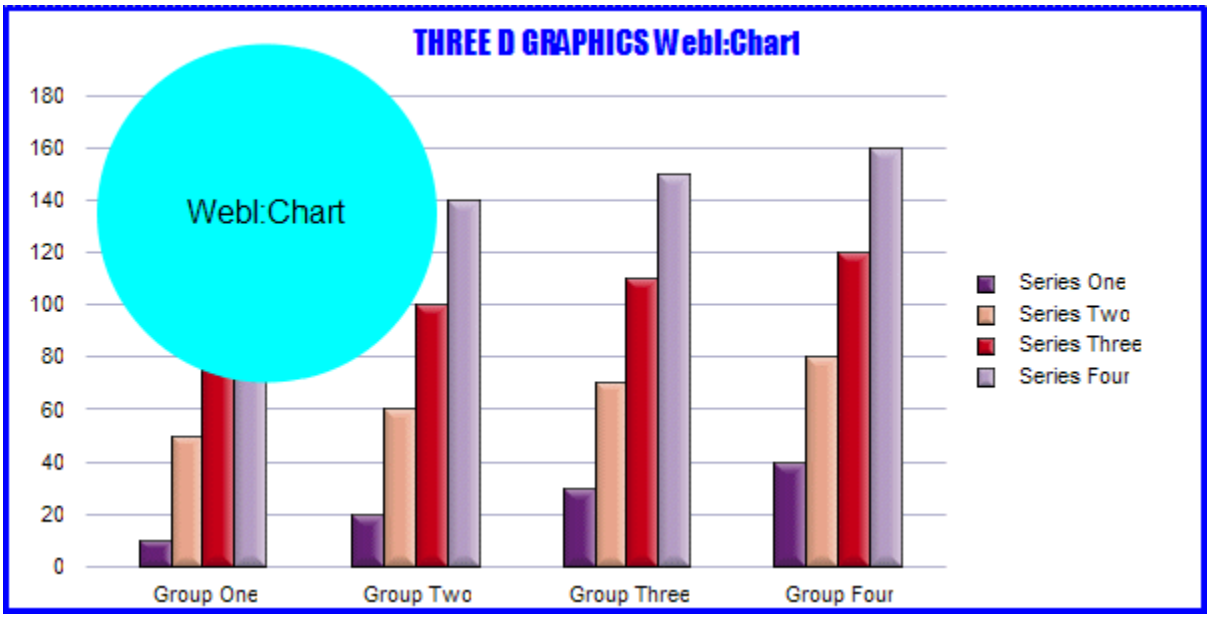
fStartY/fStopY; 0.0...1.1 Y-Axis start/stop locations

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_FILL_CIRCLE_ABOVE 0 0.0 0.4 0.4 1.1 0 255 255 WebI:Chart
```



PERSISTENT:

NO

NOTES:

- WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW.
- @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_FILL_CIRCLE2 (Pattern-Filled Circle)

On 2D charts, this macro fills a portion of the chart frame with a circle that is filled with a specified color, pattern, and optional phrase. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_FILL_CIRCLE2 fStartX fStopX fStartY fStopY nRed nGreen nBlue
nPattern szPhrase
```

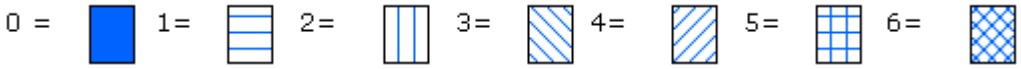
PARAMETERS:

fStartX/fStopX; 0.0...1.1 X-Axis start/stop locations

fStartY/fStopY; 0.0...1.1 Y-Axis start/stop locations

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

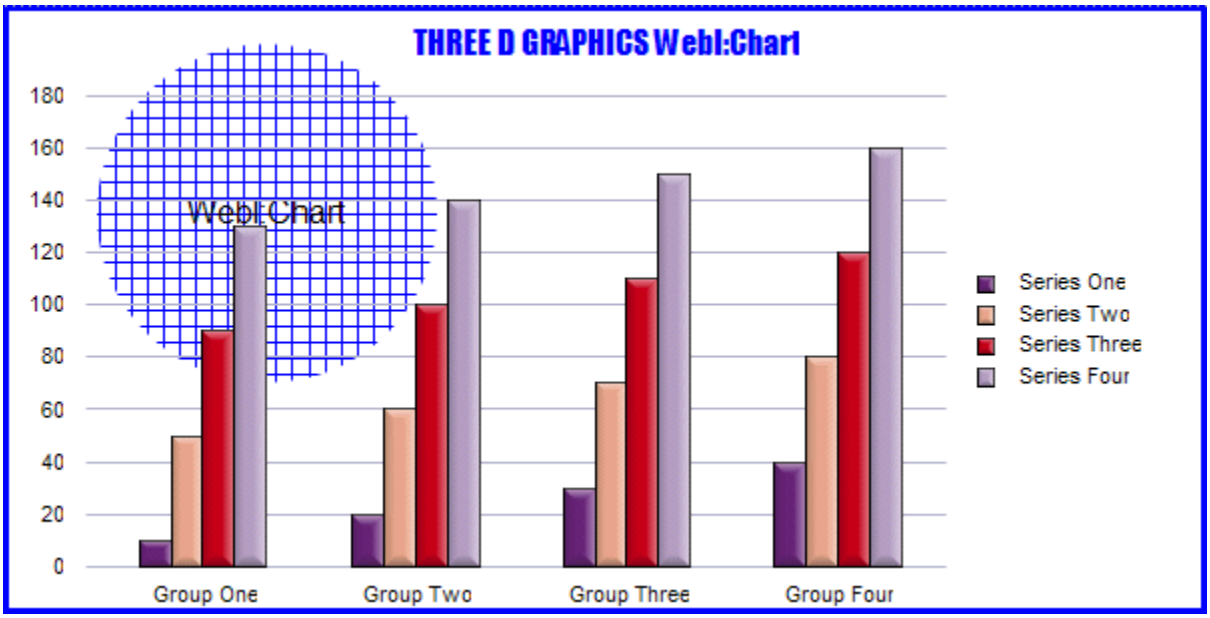
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_FILL_CIRCLE2 0 0.0 0.4 0.4 1.1 0 0 255 -5 WebI:Chart
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW. @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_FILL_CIRCLE2_ABOVE (Pattern-Filled Circle above Chart Area)

On 2D charts, this macro fills a portion of the chart frame with a circle that is filled with a specified color, pattern, and optional phrase. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame.

SYNTAX:

```
@USER_FILL_CIRCLE2_ABOVE fStartX fStopX fStartY fStopY nRed nGreen
nBlue nPattern szPhrase
```

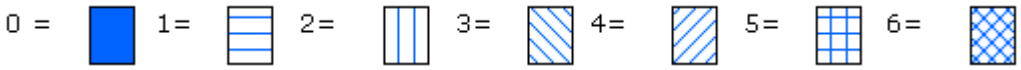
PARAMETERS:

fStartX/fStopX; 0.0...1.1 X-Axis start/stop location

fStartY/fStopY; 0.0...1.1 Y-Axis start/stop location

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

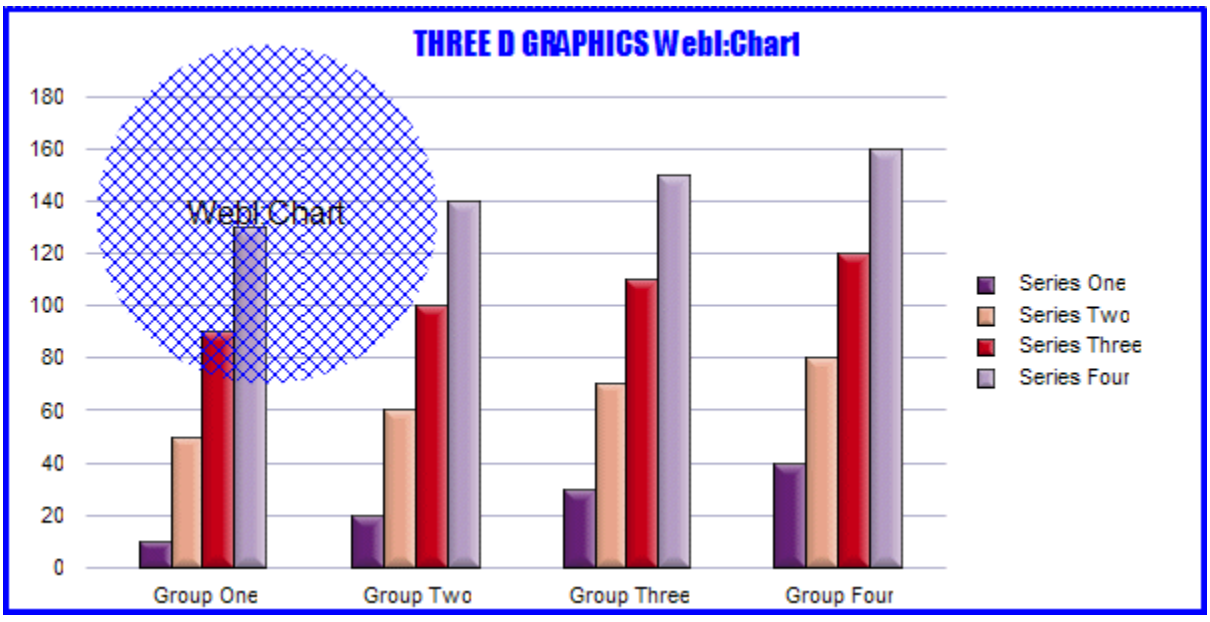
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_FILL_CIRCLE2_ABOVE 0.0 0.4 0.4 1.1 0 0 255 -6 WebI:Chart
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW. @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@USER_MARKER (User-Defined Marker)

This macro adds a user-defined marker with optional text (*szPhrase*) to a chart. *fX* must be value that is between the minimum and maximum values shown on the X-Axis. *fY* must be value that is between the minimum and maximum values shown on the Y-Axis.

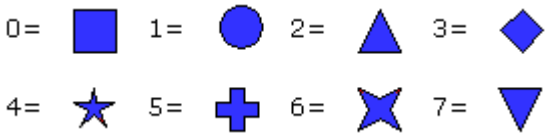
SYNTAX:

```
@USER_MARKER fX fY nShape nRed nGreen nBlue szPhrase
```

PARAMETERS:

fX; X-position, *fY:* Y-position

nShape; 0...7 selects one of the following markers:

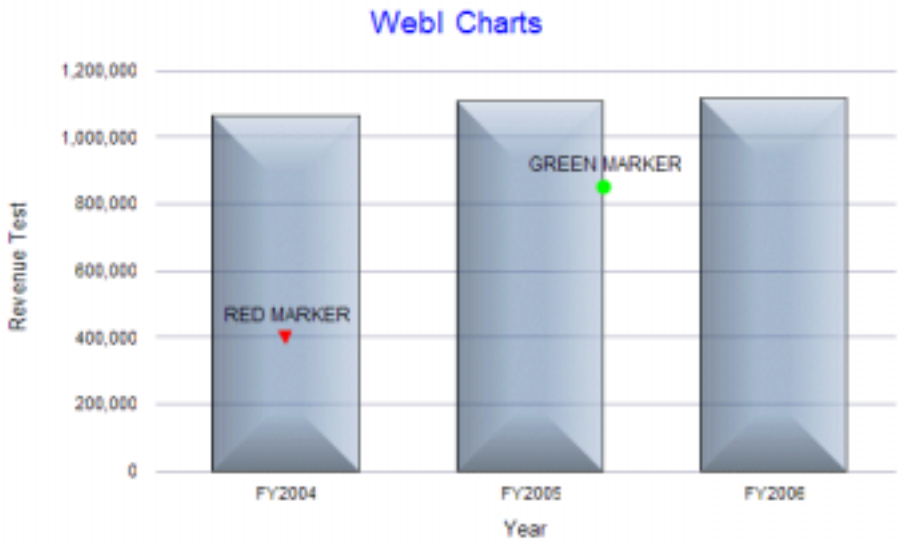


nRed; 0...255, *nGreen;* 0...255, *nBlue;* 0...255 define the color of the marker.

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_MARKER 1.5 400000 7 255 0 0 RED MARKER
@USER_MARKER 2.5 850000 1 0 255 0 GREEN MARKER
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of eight user markers defined with @USER_MARKER and @USER_MARKER2.

ALSO SEE:

@USER_MARKER2

@USER_RECT (Outlined Rectangle)

On 2D charts, this macro fills a portion of the chart frame with a rectangle. Set *fStartX* and *fStopX* to zero to select the lower left corner of the chart frame. The rectangle is outlined with the specified color (*nRed*, *nGreen*, *nBlue*). The *nThickness* and *nStyle* parameters define the thickness and style of the outline of the rectangle.

SYNTAX:

```
@USER_RECT fStartX fStopX fStartY fStopY nRed nGreen nBlue nLineStyle
nLineThickness szPhrase
```

PARAMETERS:

fStartX; 0.0...1.1 X-Axis start location

fStopX; 0.0...1.1 X-Axis stop location

fStartY; 0.0...1.1 Y-Axis start location

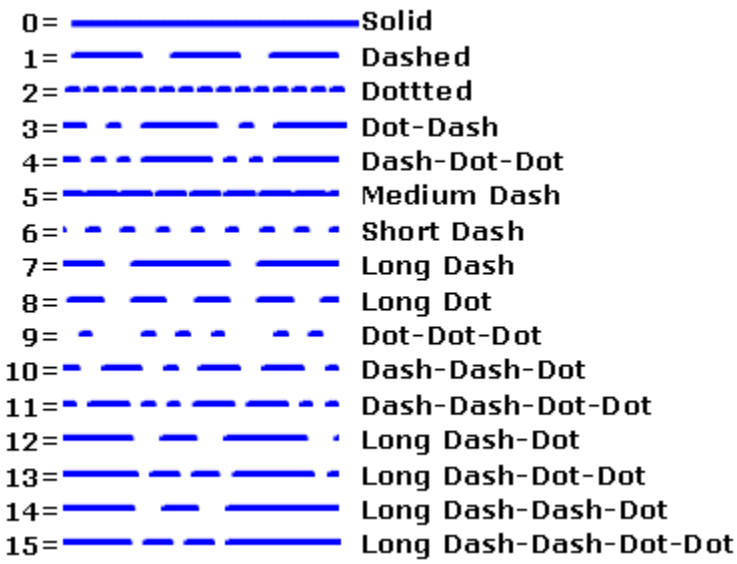
fStopY; 0.0...1.1 Y-Axis stop location

nRed; 0...255 defines the RED portion of the RGB color

nGreen; 0...255 defines the GREEN portion of the RGB color

nBlue; 0...255 defines the BLUE portion of the RGB color

nLineStyle; 0...15 selects the outline style

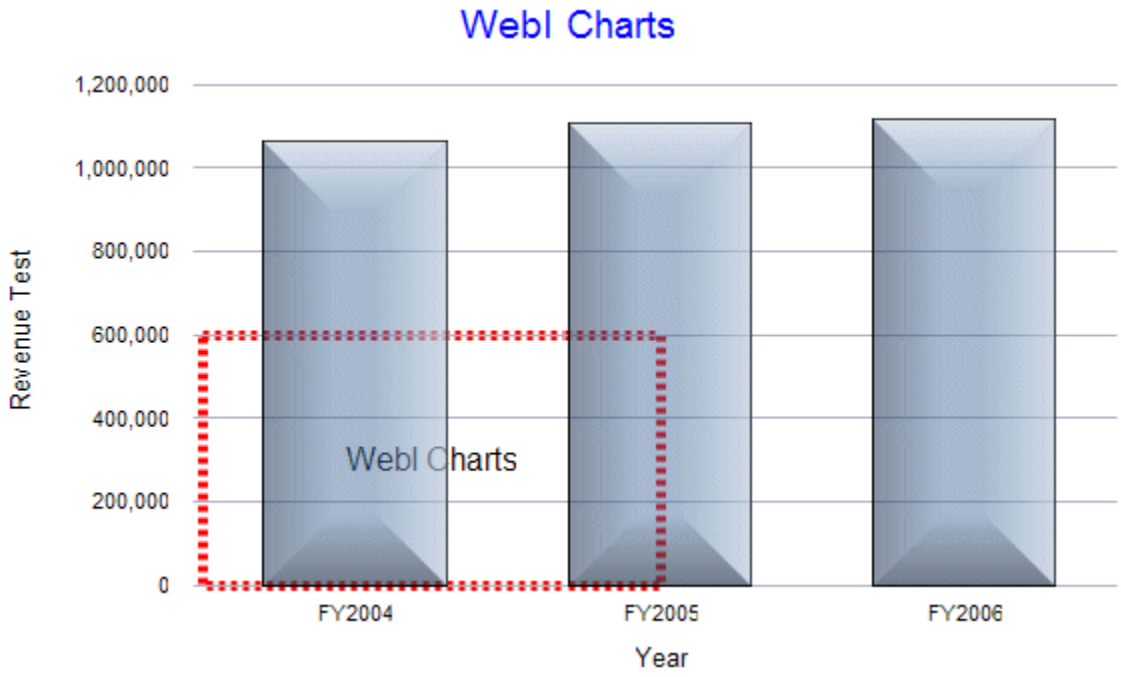


nLineThickness; 0...1000 Line Thickness

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@USER_RECT 0.0 0.5 0.0 0.5 255 0 0 2 500 WebI Charts
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW. @UF, @USER_CIRCLE..., @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.

@UW (User-Defined Vertical Band)

This macro draws a band on the chart background from *fStartX* to *fStopX* using the color defined by @WC. The default color is white.

SYNTAX:

```
@UW fStartX fStopX szPhrase
```

PARAMETERS:

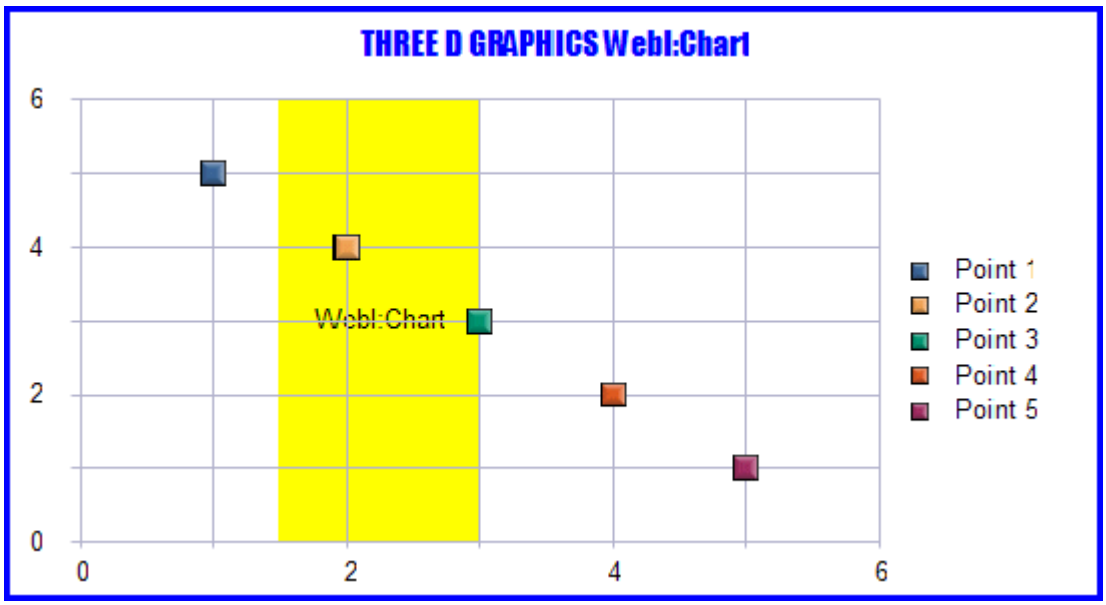
fStartX; 0.0...1.0 X-Axis start location

fStopX; 0.0...1.0 X-Axis stop location

szPhrase; Optional phrase. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@UW .25 .50 WebI:Chart
@WC 255 0 0
```



PERSISTENT:

NO

NOTES:

WebI:Chart supports a maximum of 20 user-defined areas. User-defined areas are created with: @UF, @USER_CIRCLE..., @USER_FILL..., @USER_RECT, and @UW.

ALSO SEE:

@WC

@WC (Color @UW Vertical Band)

This macro specifies the color of a vertical band defined by the @UW macro.

SYNTAX:

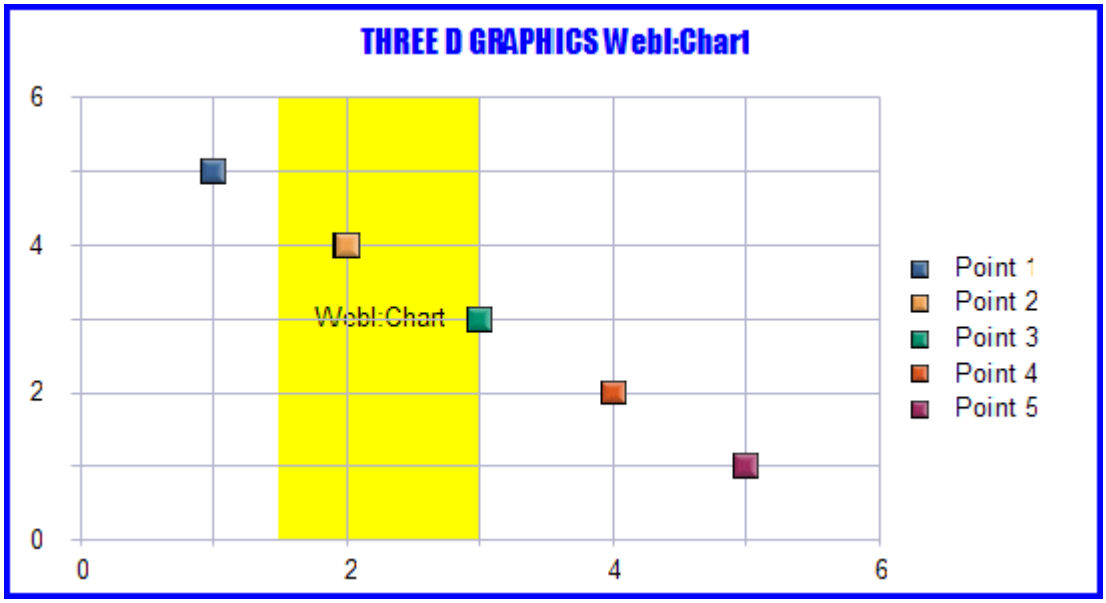
```
@WC nRed nGreen nBlue
```

PARAMETERS:

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

EXAMPLE:

```
@UW .25 .50 WebI:Chart
@WC 255 0 0
```



PERSISTENT:

NO

ALSO SEE:

@UW

Section 11: Chart Type Macros

These macros change the chart type:

- @3DSCAT; Change the chart type to a 3D Scatter Chart
- @AUDIO; Change the chart type to an Audiogram
- @COMBO; Change the chart type to a Combo Chart
- @COMPARE2; Change the chart type to a 2-Series Absolute Bar Chart
- @GRAPHTYPE; Select a Graph Type
- @PARETO; Change the chart type to a Pareto Chart
- @POLAR; Change the chart type to a Polar Chart
- @POLAR_SPIKE; Draw a line from data point to center of Polar Chart plot area
- @RIVER; Draw two series as a floating area (i.e., river)

In all cases, the data being supplied to the chart should be compatible with the selected chart type.

@3DSCAT (3D Scatter Chart)

This macro changes the chart type to a 3D Scatter chart.

SYNTAX:

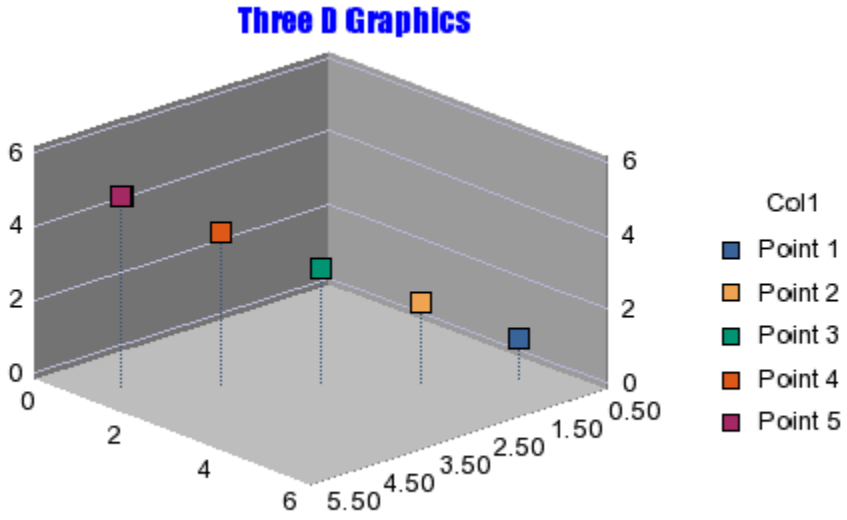
```
@3DSCAT
```

PARAMETERS:

None

EXAMPLE:

```
@3DSCAT
```



PERSISTENT:

YES

@COMBO (Combo Chart)

This macro creates a combination bar/line/area chart.

SYNTAX:

```
@COMBO nSeries nStyle
```

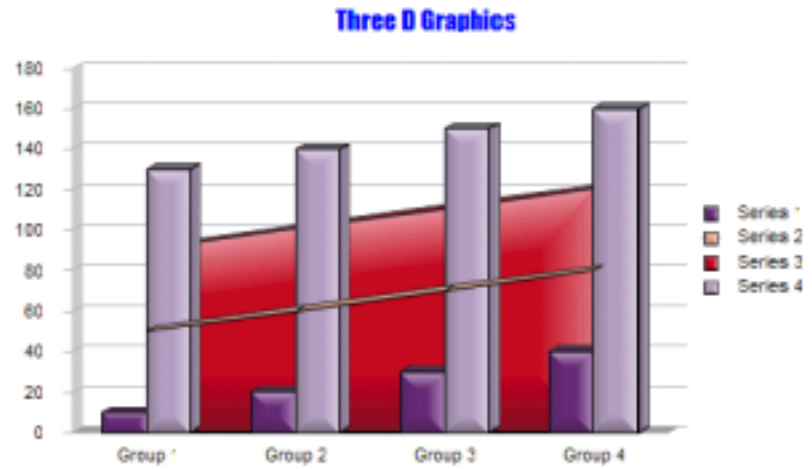
PARAMETERS:

nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

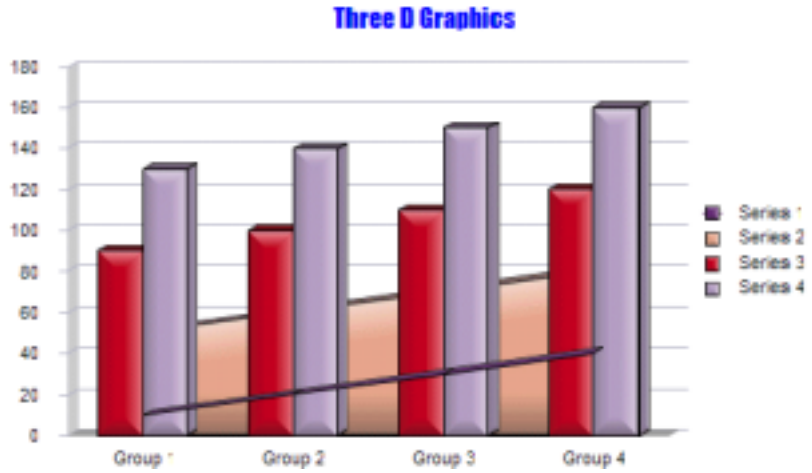
nStyle; 0...2 (0= Bar, 1=Line, 2=Area)

EXAMPLE:

```
@COMBO 0 0 @COMBO 1 1 @COMBO 2 2
```




```
@COMBO 0 1 @COMBO 1 2 @COMBO 2 0
```



PERSISTENT:

YES

@GRAPHTYPE (Graph Type)

This macro selects a different graph type and assigns it to the chart that is shown in the report.

SYNTAX:

```
@GRAPHTYPE nGraph
```

PARAMETERS:

nGraph; 0…90 selects one of the following graph types:

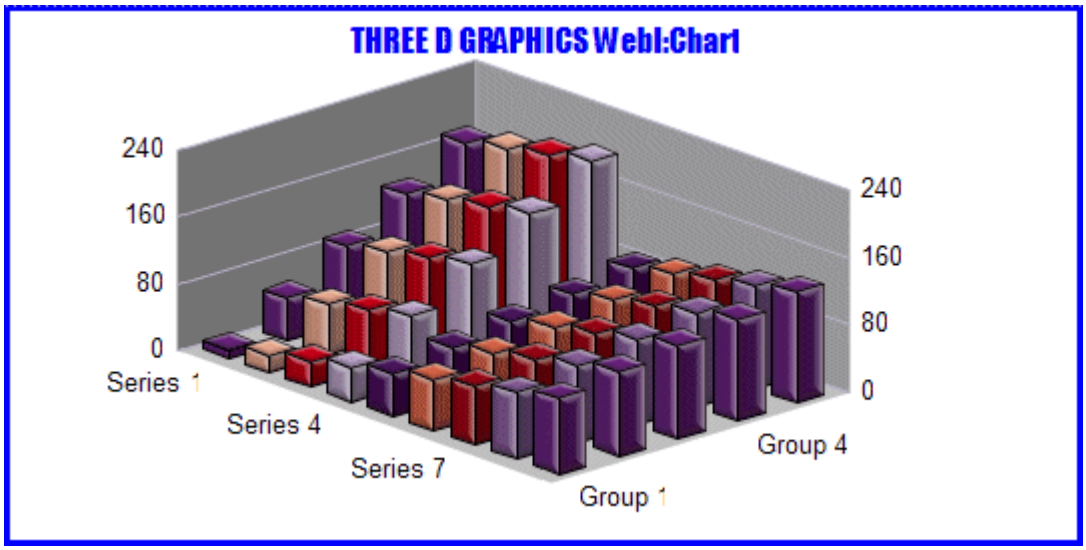
nGraph	Chart
0	Vertical Area Absolute
1	Vertical Area Stacked
2	Vertical Area Bi-Polar Absolute
3	Vertical Area Bi-Polar Stacked
4	Vertical Area Dual-Y Absolute
5	Vertical Area Dual-Y Stacked
6	Vertical Area Percent
7	Horizontal Area Absolute
8	Horizontal Area Stacked
9	Horizontal Area Bi-Polar Absolute
10	Horizontal Area Bi-Polar Stacked
11	Horizontal Area Dual-Y Absolute
12	Horizontal Area Dual-Y Stacked
13	Horizontal Area Percent
14	Vertical Bar Side-by-Side
15	Vertical Bar Stacked
16	Vertical Bar Dual-Y Side-by-Side
17	Vertical Bar Dual-Y Stacked
18	Vertical Bar Bi-Polar Side-by-Side
19	Vertical Bar Bi-Polar Stacked
20	Vertical Bar Percent
21	Horizontal Bar Side-by-Side
22	Horizontal Bar Stacked
23	Horizontal Bar Dual-Y Side-by-Side
24	Horizontal Bar Dual-Y Stacked
25	Horizontal Bar Bi-Polar Side-by-Side
26	Horizontal Bar Bi-Polar Stacked
27	Horizontal Bar Percent
28	Vertical Line Absolute
29	Vertical Line Stacked
30	Vertical Line Bi-Polar Absolute
31	Vertical Line Bi-Polar Stacked
32	Vertical Line Dual-Y Absolute
33	Vertical Line Dual-Y Stacked

nGraph	Chart
34	Vertical Line Percent
35	Horizontal Line Absolute
36	Horizontal Line Stacked
37	Horizontal Line Bi-Polar Absolute
38	Horizontal Line Bi-Polar Stacked
39	Horizontal Line Dual-Y Absolute
40	Horizontal Line Dual-Y Stacked
41	Horizontal Line Percent
42	Pie
43	Ring Pie
44	Multiple Pie
45	Multiple Ring Pies
46	Multiple Proportional Pies
47	Multiple Proportional Ring Pies
48	Pie Bar
49	Ring Pie Bar
50	X/Y Scatter
51	X/Y Scatter Dual-Y
52	X/Y Scatter with Labels
53	X/Y Scatter Dual-Y with Labels
54	Polar
55	Polar Dual-Y
56	Radar
57	Radar Stacked
58	Radar Dual-Y
59	Radar Stacked Dual-Y
60	Bubble
61	Bubble Dual-Y
62	Gantt
63	High-Low Stock Chart
64	High-Low Dual-Y Stock Chart
65	High-Low-Open Stock Chart
66	High-Low-Open Dual-Y Stock Chart
67	High-Low-Open-Close Stock Chart
68	High-Low-Open-Close Dual-Y Stock Chart
69	Spectral Map
70	Vertical Histogram
71	Horizontal Histogram
72	Table
73	3D Riser – Bars
74	3D Riser – Pyramids

nGraph	Chart
75	3D Riser – Octagons
76	3D Riser – Cut-Corner Bars
77	3D Floating – Cubes
78	3D Floating – Spheres
79	3D Connect Group – Areas
80	3D Connect Group – Ribbons
81	3D Connect Group – Steps
82	3D Connect Series – Areas
83	3D Connect Series – Ribbons
84	3D Connect Series – Steps
85	3D Surface
86	3D Surface with Sides
87	3D Honeycomb Surface
88	3D X/Y/Z Scatter Chart
89	3D X/Y/Z Scatter Chart with Labels
90	Box Plot

EXAMPLE:

```
@GRAPHTYPE  73
```



PERSISTENT:

YES

@PARETO (Pareto Chart)

This macro creates a Pareto chart. If *bType* is zero, a simple Pareto chart is created. If *bType* is one, a Classic Pareto chart with a cumulative percentage line is created.

SYNTAX:

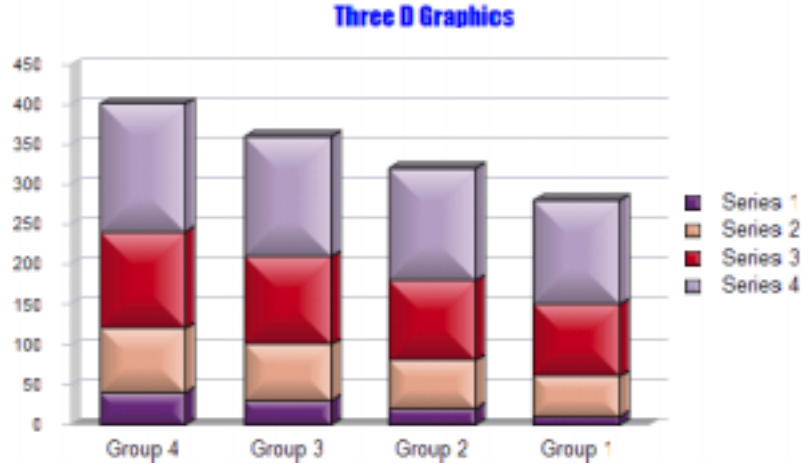
```
@PARETO bType
```

PARAMETERS:

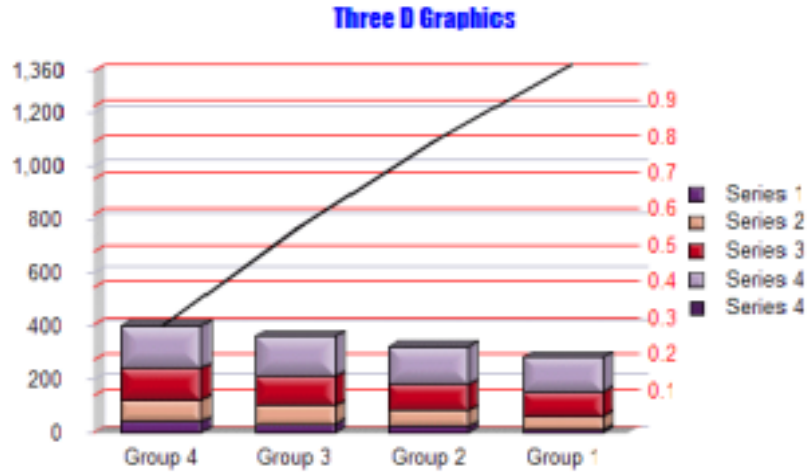
bType; 0 = create a simple Pareto chart. 1 = create a Classic Pareto chart with a cumulative percentage line.

EXAMPLE:

```
@PARETO 0
```




```
@PARETO 1
```



PERSISTENT:

YES

@POLAR (Polar Chart)

This macro changes the chart type to a Polar (circular scatter) chart.

SYNTAX:

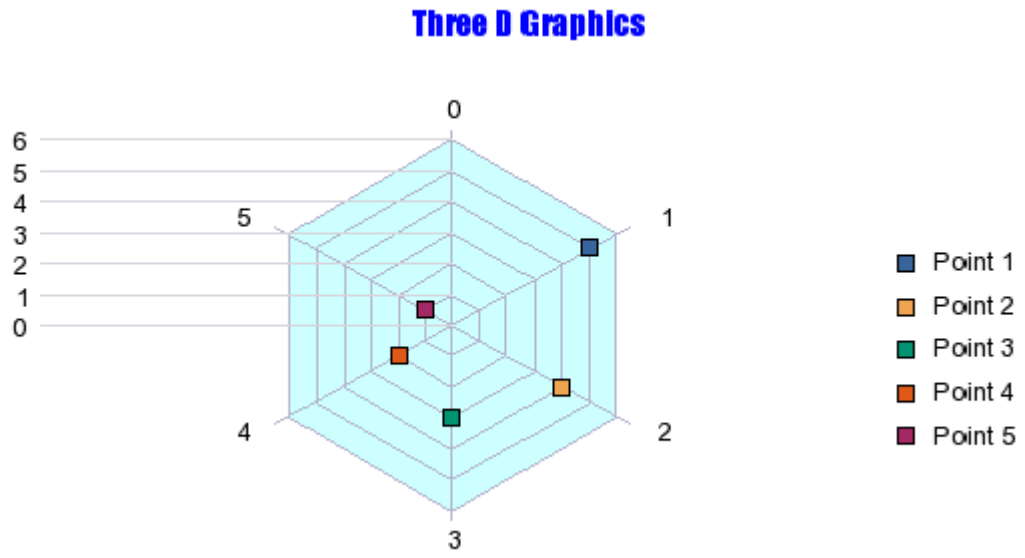
```
@POLAR
```

PARAMETERS:

None

EXAMPLE:

```
@POLAR
```



PERSISTENT:

YES

@POLAR_SPIKE (Polar Spike Chart)

This macro draws a line from data points to the center of a Polar Chart plot area.

SYNTAX:

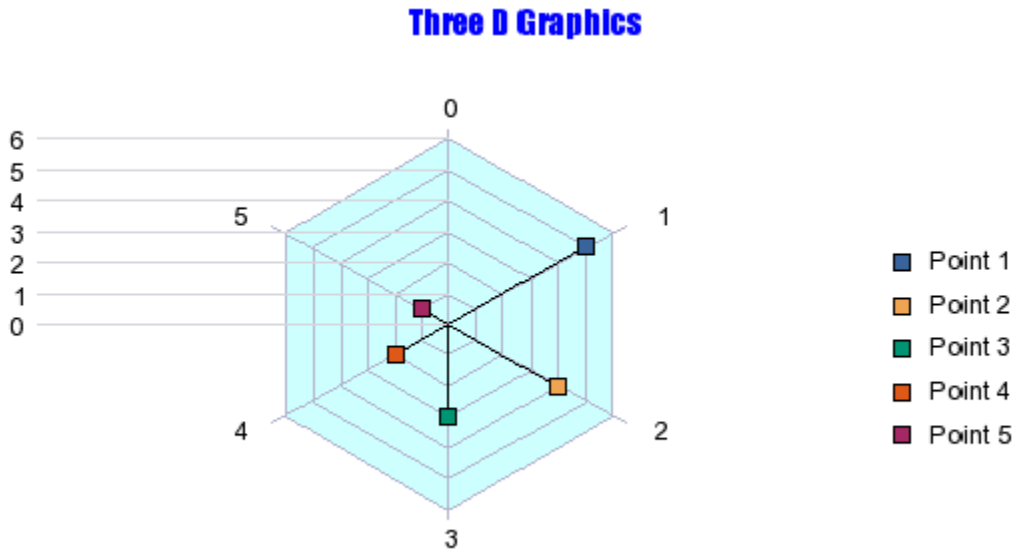
```
@POLAR_SPIKE bSpike
```

PARAMETERS:

bSpike; 0/1. 0 = Disable lines, 1 = Enables lines.

EXAMPLE:

```
@POLAR_SPIKE 1
```



PERSISTENT:

NO


WebI: CHART

Section 12: Box Plot Macros

These macros create and modify box plot charts:

- @BP/BP1; Create a box plot chart with a square-style tail
- @BP2; Create a box plot chart with a T-style tail
- @BP3; Create a box plot chart with a I-style tail
- @BPH; Change the orientation of a Box Plot chart
- @BPW; Use the sixth data value to determine the width of Box Plots
- @MC; Define Marker Colors in Box Plots
- @MK; Define the Number of Markers in Box Plots
- @MS; Define Marker Shapes in Box Plots

Also see the @SZ macro in Section 6 (Risers & Markers) to set the size of markers in box plots.

@BP/BP1 (Box Plot with Square Tail)

These macros change the chart type to a Box Plot with a square tail (standard). Each box in a Box Plot requires five values. Each set of five values form the "box" and define the location of the markers on top of each box.

SYNTAX:

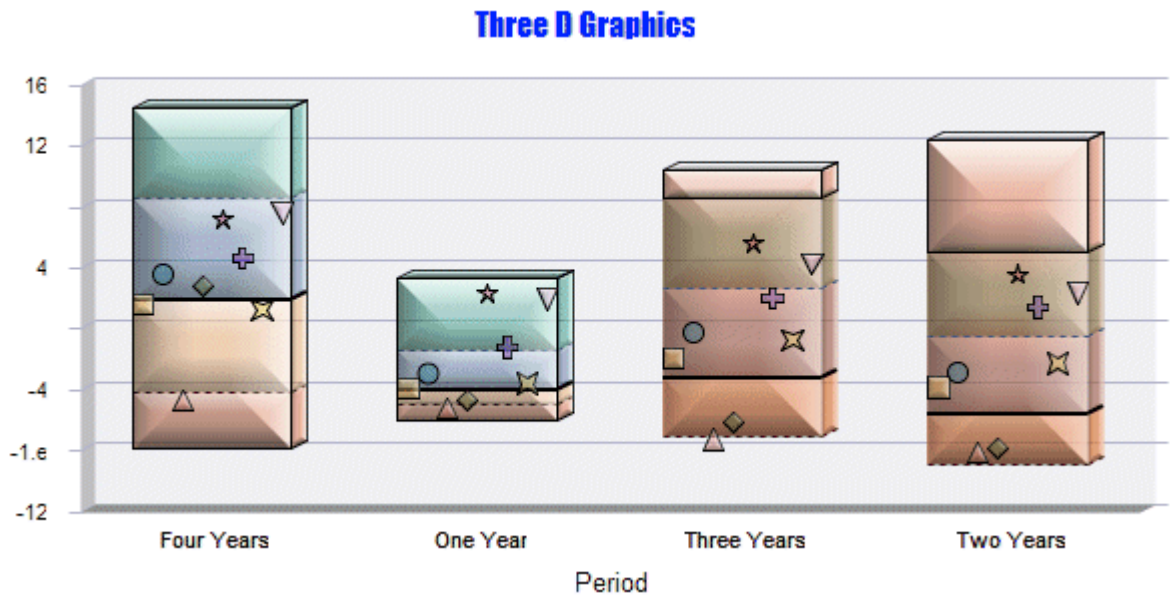
```
@BP or @BP1
```

PARAMETERS:

None

EXAMPLE:

```
@BP
```



PERSISTENT:

NO

ALSO SEE:

@BP2, @BP3

NOTES:

The default marker shape for markers on box plots are:

These default marker shapes can be changed using the "@MS" macro.

@BP2 (Box Plot with T-Style Tail)

This macro changes the chart type to a Box Plot with a T-Style tail.

SYNTAX:

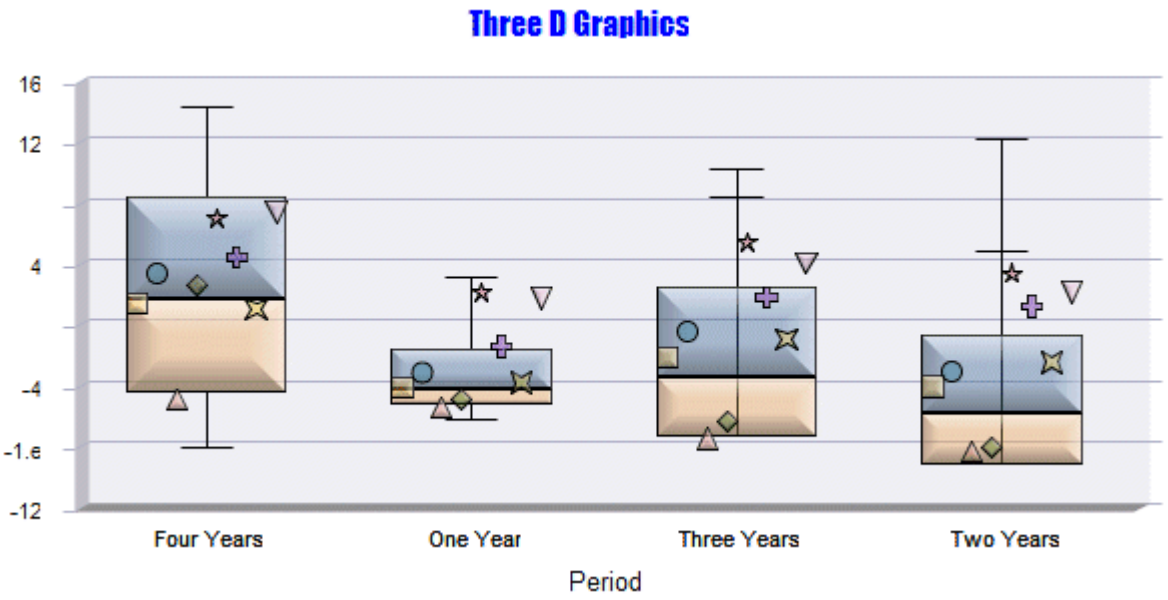
```
@BP2
```

PARAMETERS:

None

EXAMPLE:

```
@BP2
```



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP3

NOTES:

The default marker shape for markers on box plots are:

These default marker shapes can be changed using the “@MS” macro.

@BP3 (Box Plot with I-Style Tail)

This macro changes the chart type to a Box Plot with an I-Style tail.

SYNTAX:

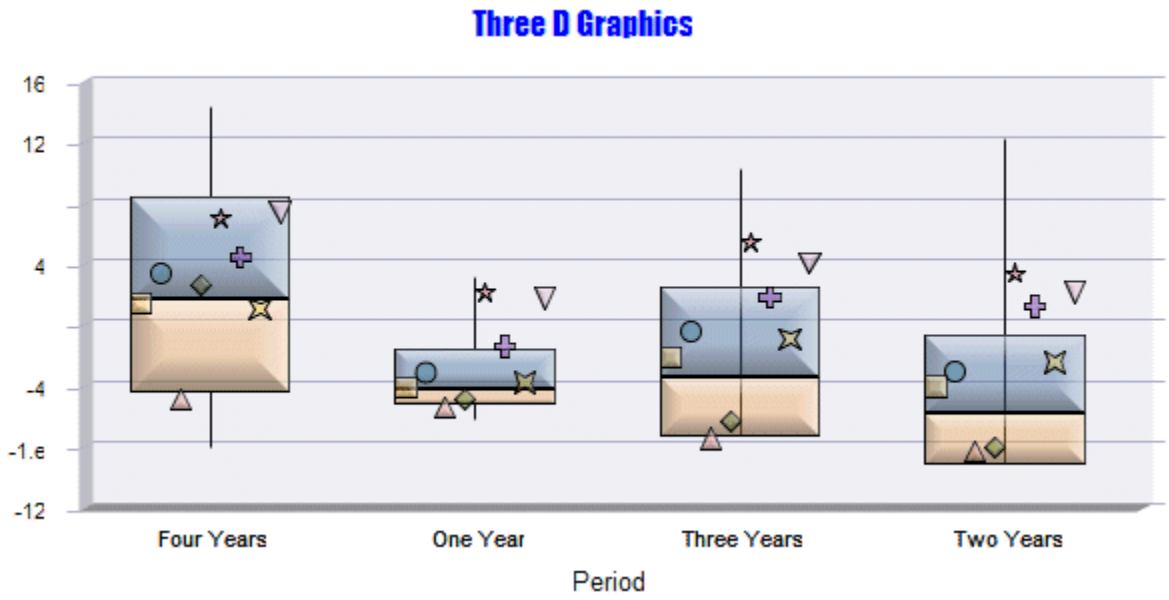
```
@BP3
```

PARAMETERS:

None

EXAMPLE:

```
@BP3
```



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP2

NOTES:

The default marker shape for markers on box plots are:

These default marker shapes can be changed using the “@MS” macro.

@BPH (Horizontal Box Plot)

This macro can be used to change the orientation of a Box Plot chart.

SYNTAX:

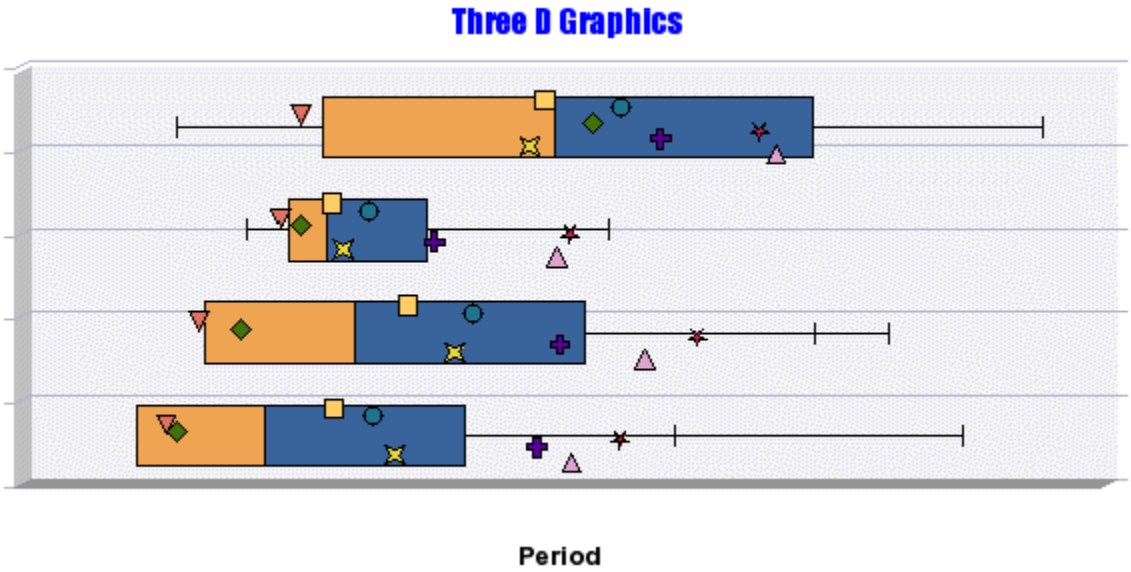
```
@BPH bOrient
```

PARAMETERS:

bOrient; 0 = Vertical Box Plot, 1= Horizontal Box Plot

EXAMPLE:

```
@BP2 @SZ 24 @BPH 1
```



PERSISTENT:

NO

@BPW (Box Plot Width)

When this macro is used in a box plot chart, the width of each box plot riser is determined by a sixth data value. The largest value gets 100% (normal) width. The width of all other risers (box plots) is a percentage of the largest value. Note that this macro is the same as @BP except that it causes the special draw with width being determined by a 6th variable.

SYNTAX:

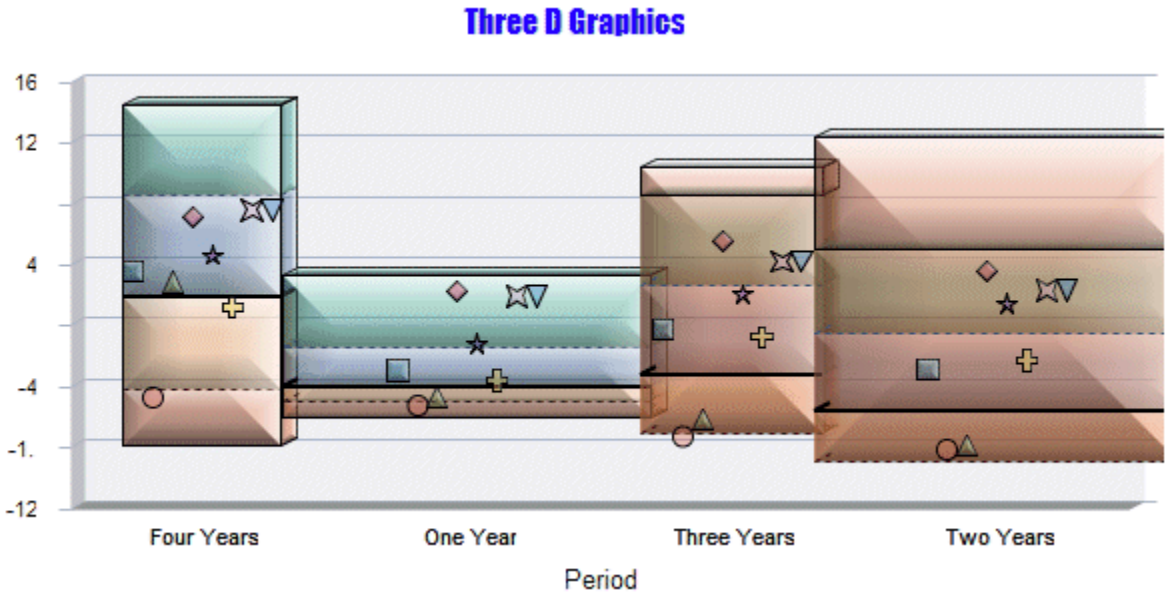
```
@BPW
```

PARAMETERS:

None

EXAMPLE:

```
@BPW
```



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP2, @BP3

@BW (Black/White Box Plot)

This macro creates a black/white Box Plot.

SYNTAX:

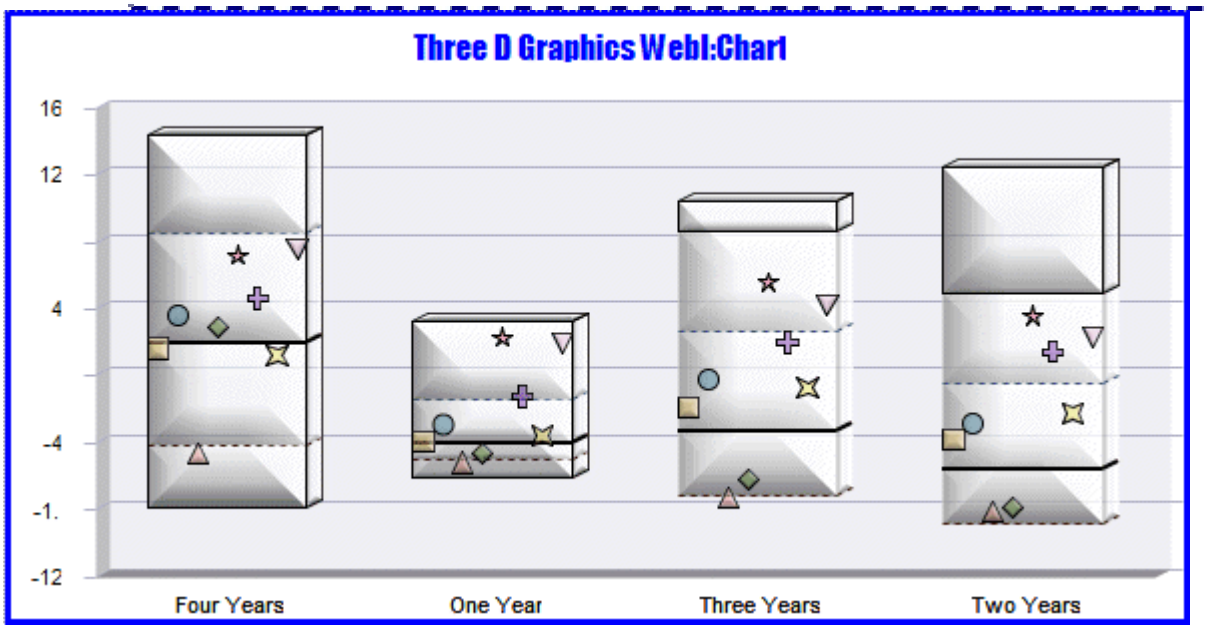
```
@BW
```

PARAMETERS:

None

EXAMPLE:

```
Three D Graphics WebI:Chart~ @FONTNAME 1 2 @BP @SZ 35 @HQ 1
@BEVEL 3 1 @ALPHA -1 100 @BW
```



PERSISTENT:

NO

@MC (Marker Colors for Box Plots)

For Box Plots only, this macro sets the color of a particular series in the chart.

SYNTAX:

```
@MC nSeries nRed nGreen nBlue
```

PARAMETERS:

nSeries; Series Number (1...8)

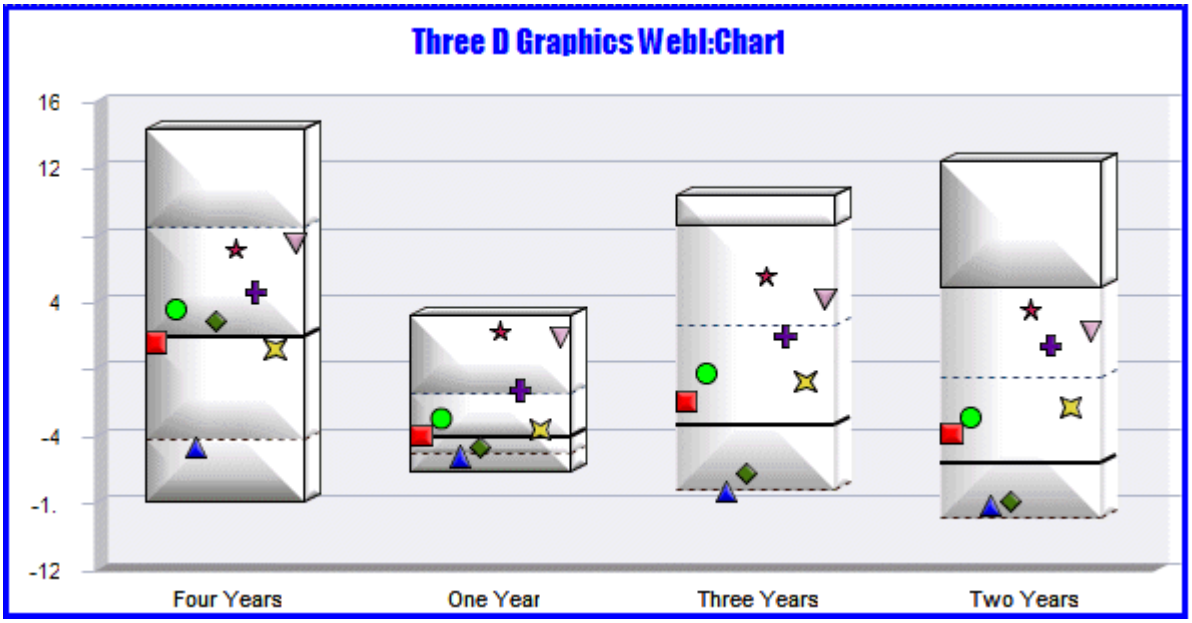
nRed; 0...255 defines the Red portion of RGB color selection.

nGreen; 0...255 defines the Green portion of RGB color selection.

nBlue; 0...255 defines the Blue portion of RGB color selection.

EXAMPLE:

```
Three D Graphics WebI:Chart~ @FONTNAME 1 2 @BP @SZ 35 @HQ 1
@BEVEL 3 1 @BW
@MC 1 255 0 0 @MC 2 0 255 0 @MC 3 0 0 255
```



PERSISTENT:

YES

ALSO SEE:

@GCOLOR to change the color of other chart objects.

NOTES:

This macro is for box plot charts only, Use the @MCOLOR macro to change the color of markers and risers in other chart types.

@MK (Number of Markers)

This macro sets the number of markers to be created on top of a Box Plot.

SYNTAX:

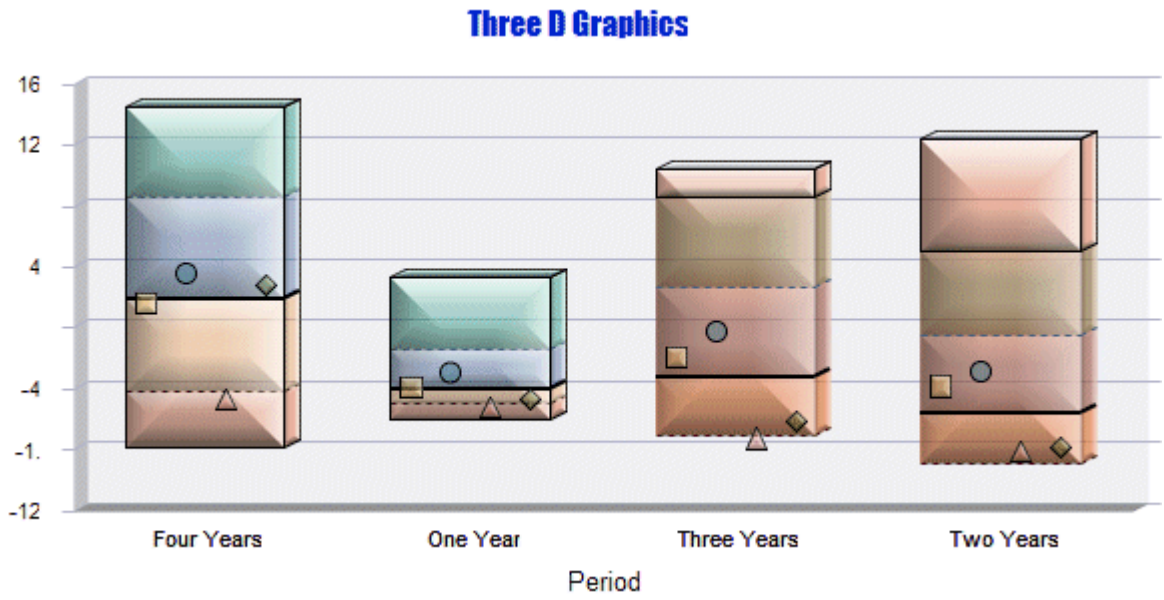
```
@MK nMarkers
```

PARAMETERS:

nMarkers; Number of markers (1...8)

EXAMPLE:

```
@BP @MK 4
```



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP2, @BP3, @MS

NOTES:

The default marker shape for markers on box plots are:

These default marker shapes can be changed using the “@MS” macro.

@MS (Marker Shapes for Box Plots)

For Box Plots only, this macro sets the shape of markers for a particular series in a chart.

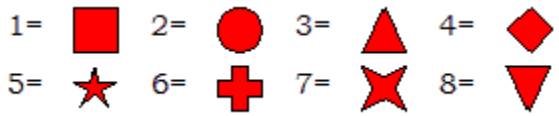
SYNTAX:

```
@MS nSeries nMarker
```

PARAMETERS:

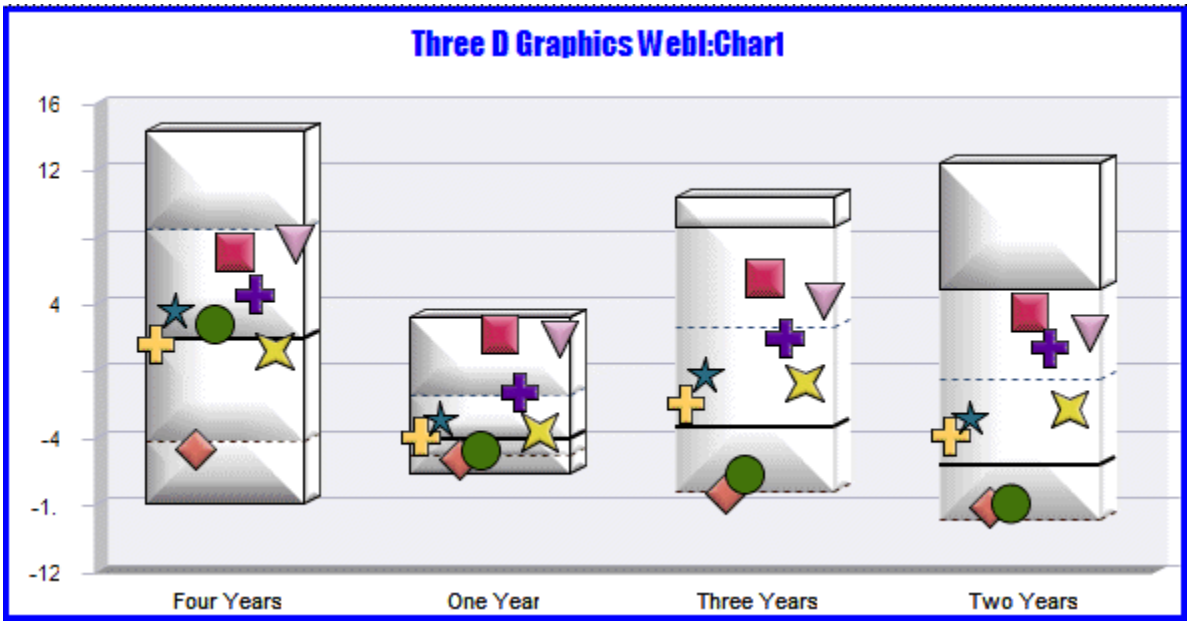
nSeries; Series Number (1...8).

nMarker; 1...8 selects the marker to assign to series *nSeries*:



EXAMPLE:

```
Three D Graphics WebI:Chart~ @FONTNAME 1 2 @BP @SZ 80 @HQ 1
@BEVEL 3 1 @BW
@MS 1 6 @MS 2 5 @MS 3 4 @MS 4 3 @MS 5 1
```



PERSISTENT:

NO

NOTES:

This macro is for box plot charts only. Use the @MARKER macro to select marker shapes in other chart types.

ALSO SEE:

- @SZ to set the size of markers in box plots
- @MC to change the color of markers in box plots

Section 13: Gauge Macros

These macros can be used to format and control the appearance of gauges.

- @GAUGE; Create a gauge chart
- @GAUGE_BORDER_STYLE; Select gauge border
- @GAUGE_BORDER_THICKNESS; Control the thickness of the new gauge engine
- @GAUGE_COLOR; Control the color of any band on the gauge
- @GAUGE_MULTIPLE_NEEDLES; Draw all or only first series as a gauge needle
- @GAUGE_PER_ROW; Number of gauges per row in a multi-gauge chart
- @GAUGE_RANGE; Gauge range colors
- @GAUGE_STARTSTOP; Gauge axis start/stop angle
- @GAUGE_STYLE; Enable/Disable new gauge drawing engine
- @GAUGE_THRESHOLD; Define Gauge Band Thresholds

@GAUGE (Gauge Chart)

This macro creates Gauge Chart.

SYNTAX:

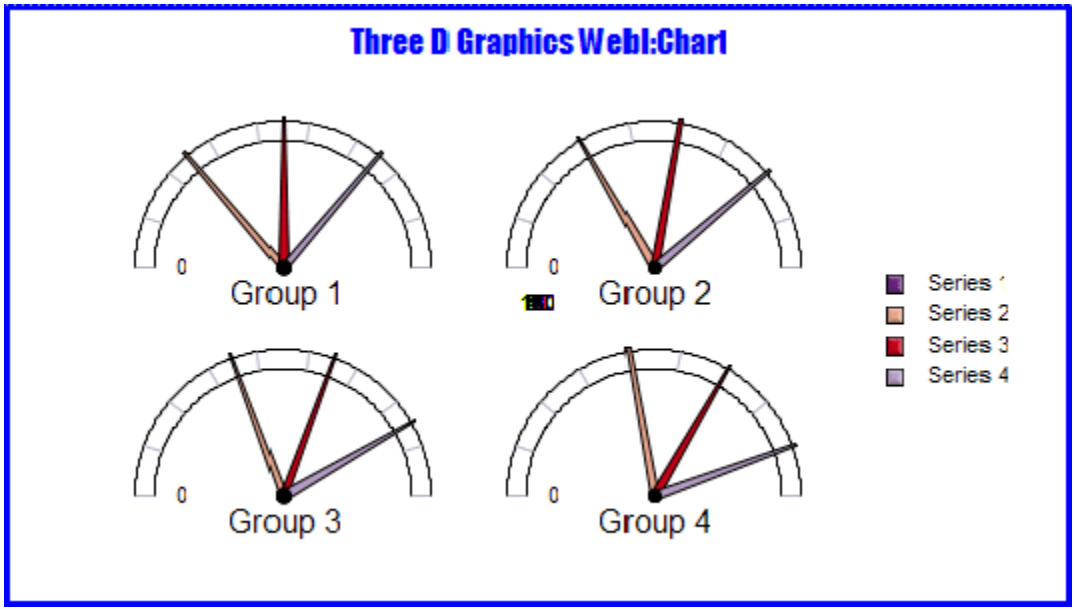
```
@GAUGE
```

PARAMETERS:

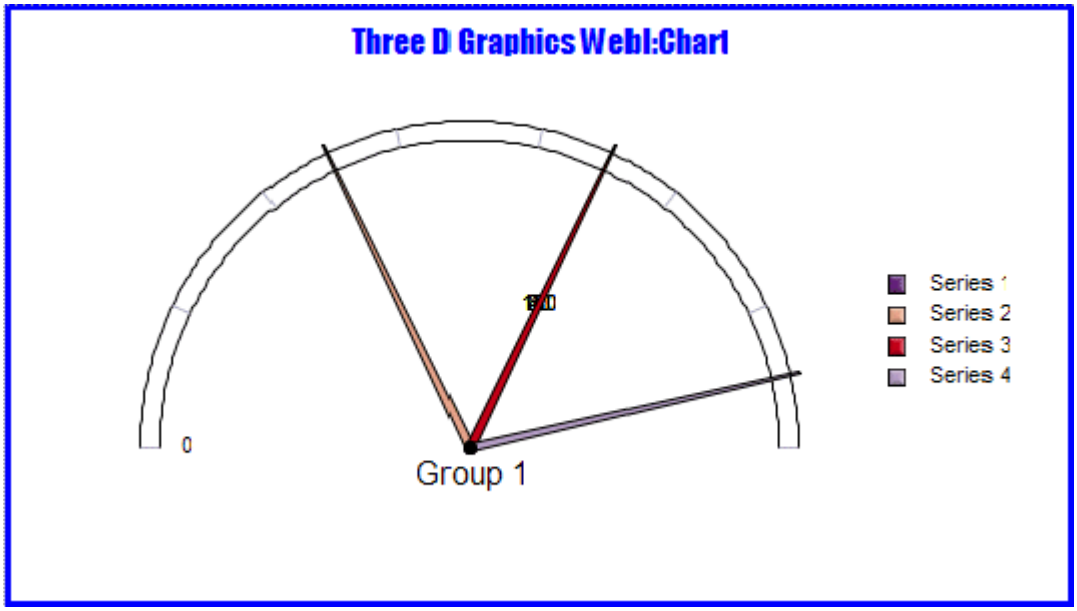
None.

EXAMPLE:

```
@GAUGE
```




```
@GAUGE @IG 1 @IG 2 @IG 3
```



PERSISTENT:

NO

@GAUGE_BORDER_STYLE (Gauge Border Style)

When gauge style is enabled (@GAUGE_STYLE 1), this macro can be used to select a gauge border style.

SYNTAX:

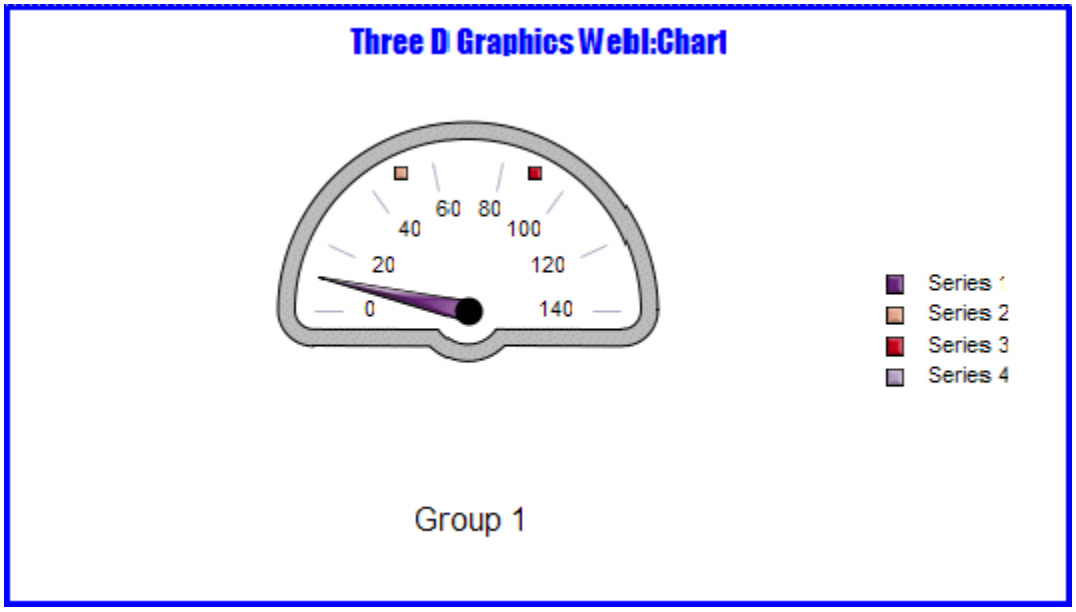
```
@GAUGE_BORDER_STYLE nStyle
```

PARAMETERS:

nStyle; 0...6 selects a border style. 0=None, 1=Simple Round, 2=3D, 3=Embossed, 4=Donut, 5=Metallic, 6=Clipped.

Example:

```
@GAUGE_BORDER_STYLE 6
```



PERSISTENT:

NO

@GAUGE_BORDER_THICKNESS (Gauge Border Thickness)

When gauge style is enabled (@GAUGE_STYLE 1), this macro controls the thickness of the gauge border.

SYNTAX:

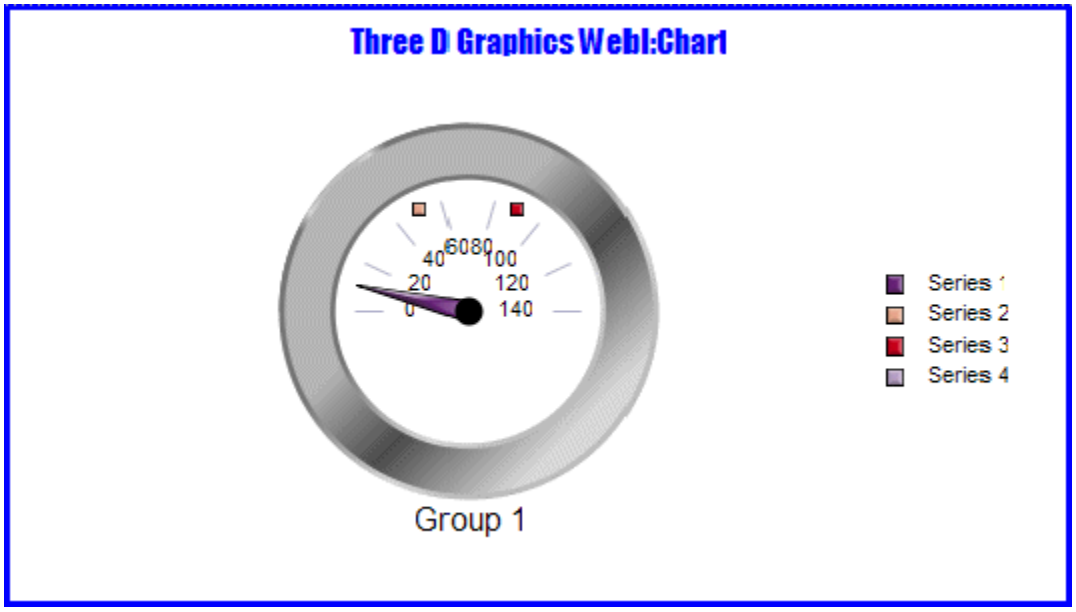
```
@GAUGE_BORDER_THICKNESS nThick
```

PARAMETERS:

nThick; 0...100 selects the thickness of the border, as a percentage of the overall possible thickness. 0=Single Line, 100=All Border.

Example:

```
@GAUGE_BORDER_THICKNESS 50
```



PERSISTENT:

NO

@GAUGE_COLOR (Color Gauge Bands)

This macro can be used to apply a different color to each of the gauge bands.

SYNTAX:

```
@GAUGE_COLOR nBand nRed nGreen nBlue
```

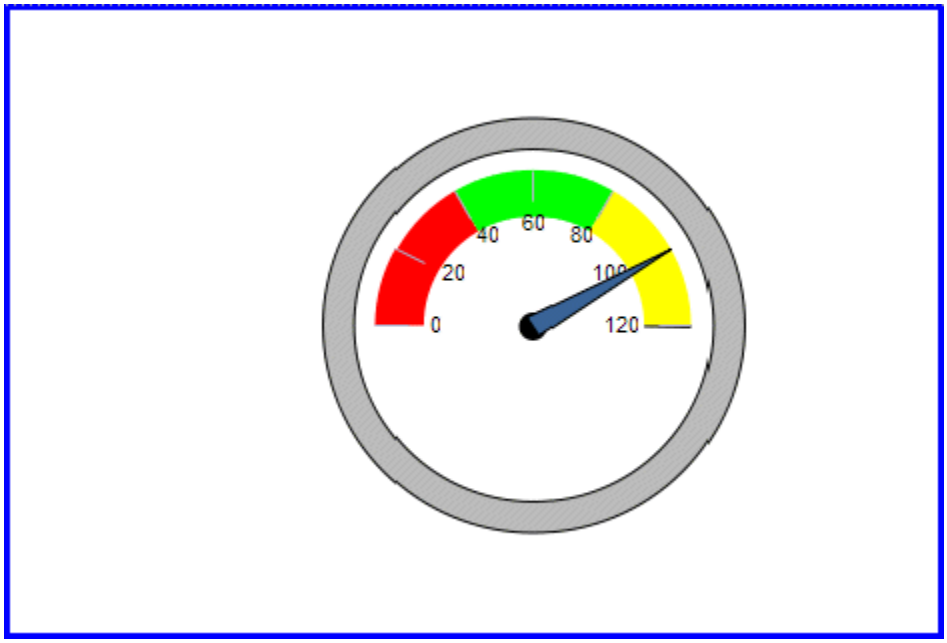
PARAMETERS:

nBand; 0...5 selects the color band.

nRed, nGreen, nBlue; 0...255 color to use for *nBand*.

EXAMPLE:

```
@GAUGE_THRESHOLD 0 40
@GAUGE_COLOR 0 255 0 0
@GAUGE_THRESHOLD 1 80
@GAUGE_COLOR 1 0 255 0
@GAUGE_THRESHOLD 2 120
@GAUGE_COLOR 2 255 255 0
```



PERSISTENT:

NO

@GAUGE_MULTIPLE_NEEDLES (Multiple Needles)

When gauge style is enabled (@GAUGE_STYLE 1) and a gauge chart includes more than one series, this macro controls the number of needles.

SYNTAX:

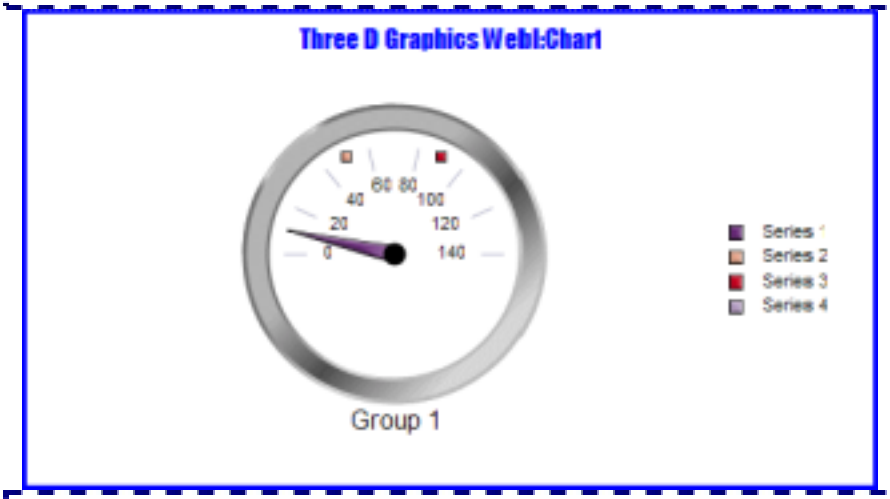
```
@GAUGE_MULTIPLE_NEEDLES bEnable
```

PARAMETERS:

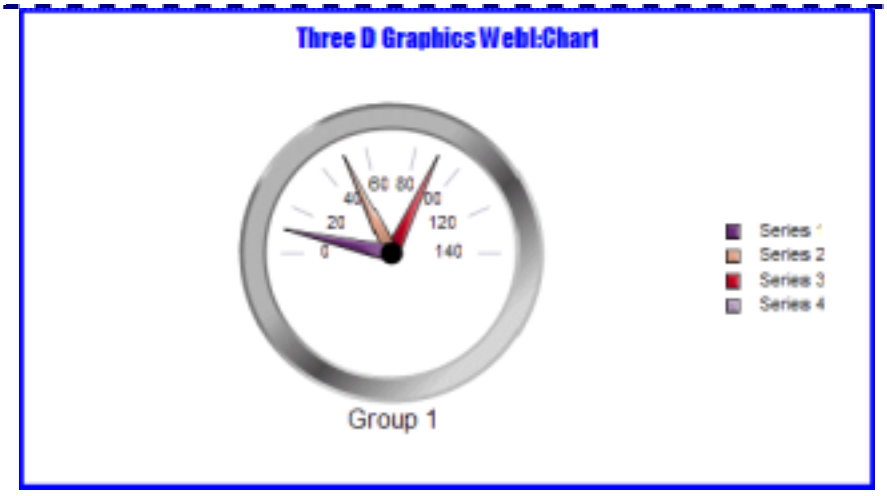
bEnable; 0=Draw first series as a needle, draw all other series as markers. 1=Draw all series as needles.

EXAMPLE:

```
@GAUGE_STYLE 1
@GAUGE_MULTIPLE_NEEDLES 0
```




```
@GAUGE_STYLE 1
@GAUGE_MULTIPLE_NEEDLES 1
```



PERSISTENT:

NO

@GAUGE_PER_ROW (Gauges Per Row)

For charts that include multiple gauges, the macros determines the number of gauges to draw in a horizontal row.

SYNTAX:

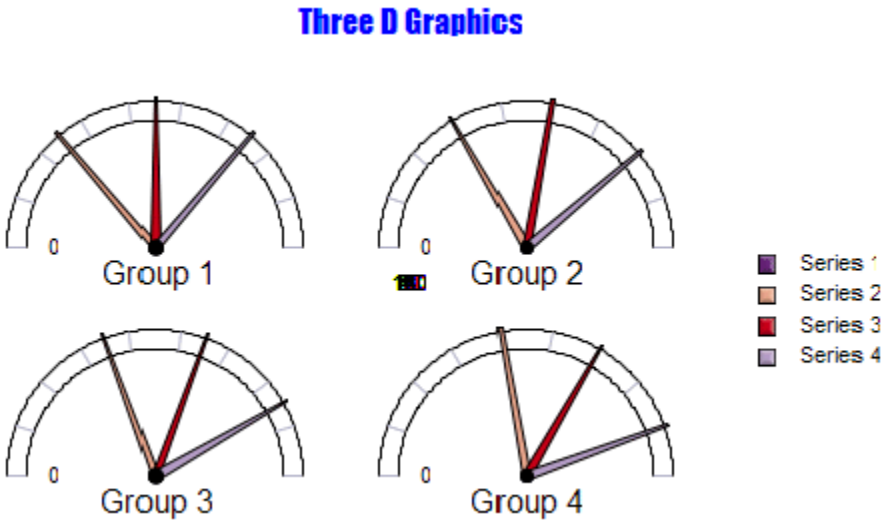
```
@GAUGE_PER_ROW nGauges
```

PARAMETERS:

nGauges; 1...# Groups in chart

Example:

```
@GAUGE @GAUGE_PER_ROW 2
```



@GAUGE_RANGE (Gauges Range Colors)

defines up to 5 color ranges that are displayed on the scale

SYNTAX:

```
@GAUGE_RANGE number from_value to_value fill
```

PARAMETERS:

number; 0, 1, 2, 3, 4 or 5

from_value; start value of range

to_value; end value of range

fill; Color, Gradient, or Image defined in any of the following formats:

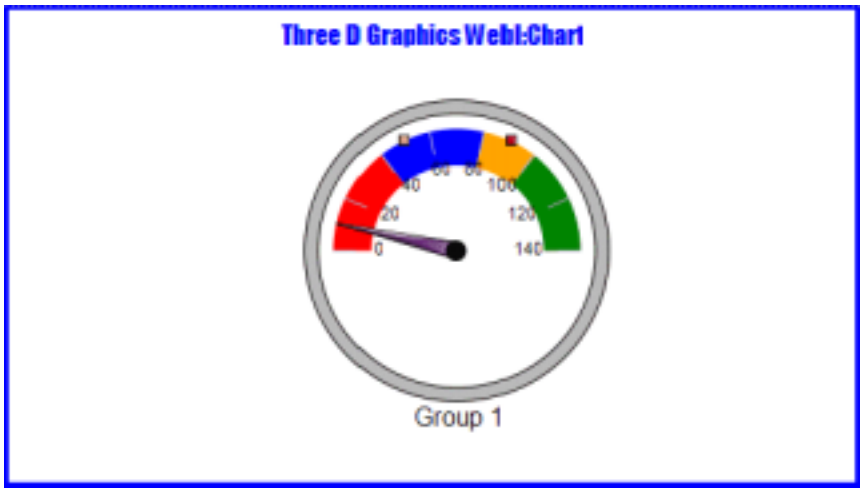
	Data Type	Range	Example
Color	Hex	0x000000...0xFFFFFF	0xFF0000 (Red)
	HTML	#000000...#FFFFFF	#FF0000 = Red
	Integer	three integers 0...255 specifying the red, blue, & green components of the color	255 0 0 = Red
	String	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red
Color + Transparency	String: Integer	a color name string from http://www.w3schools.com/CSS/css_colornames.asp	Red:224 (almost opaque)

	HTML	#00000000...#FFFFFFE0 where: last two characters specify transparency level	#ff0000e0
			#ff0000=Red
			e0=Transparency

Gradient	String	a gradient name defined by the @GRAD or @ GRAD_ABA macro	@GAUGE_RANGE 0 0 10 MyGradientDef
Picture	String	a gradient name defined by the @PICT or @PICTF macro	@GAUGE_RANGE 0 0 10 MyPictureDef

EXAMPLE:

```
@GAUGE_RANGE 0 0 40 Red
@GAUGE_RANGE 1 40 80 Blue
@GAUGE_RANGE 2 80 100 Orange
@GAUGE_RANGE 3 100 140 Green
```



@GAUGE_STARTSTOP (Gauge Start/Stop Locations)

This macro sets the start and stop locations of the gauge axis.

SYNTAX:

```
@GAUGE_STARTSTOP nStart nStop
```

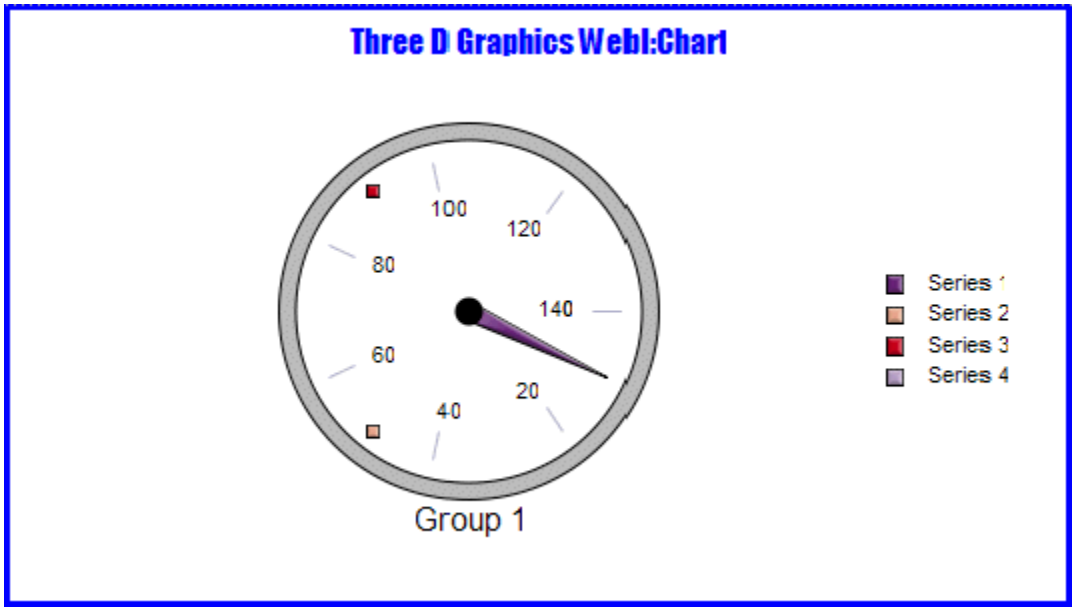
PARAMETERS:

nStart; 0...360 start location

nStop; 0...360 stop location

EXAMPLE:

```
@GAUGE_STARTSTOP 0 360
```



PERSISTENT:

NO

@GAUGE_STYLE (Gauge Style)

This macro enables/disables the enhanced gauge drawing engine.

SYNTAX:

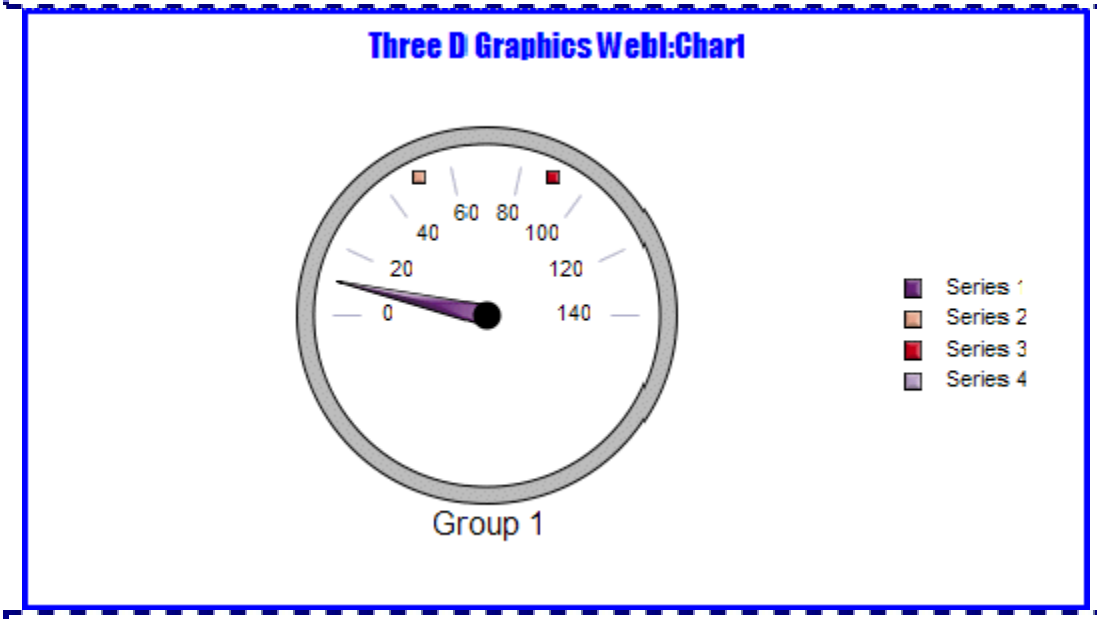
```
@GAUGE_STYLE bEnable
```

PARAMETERS:

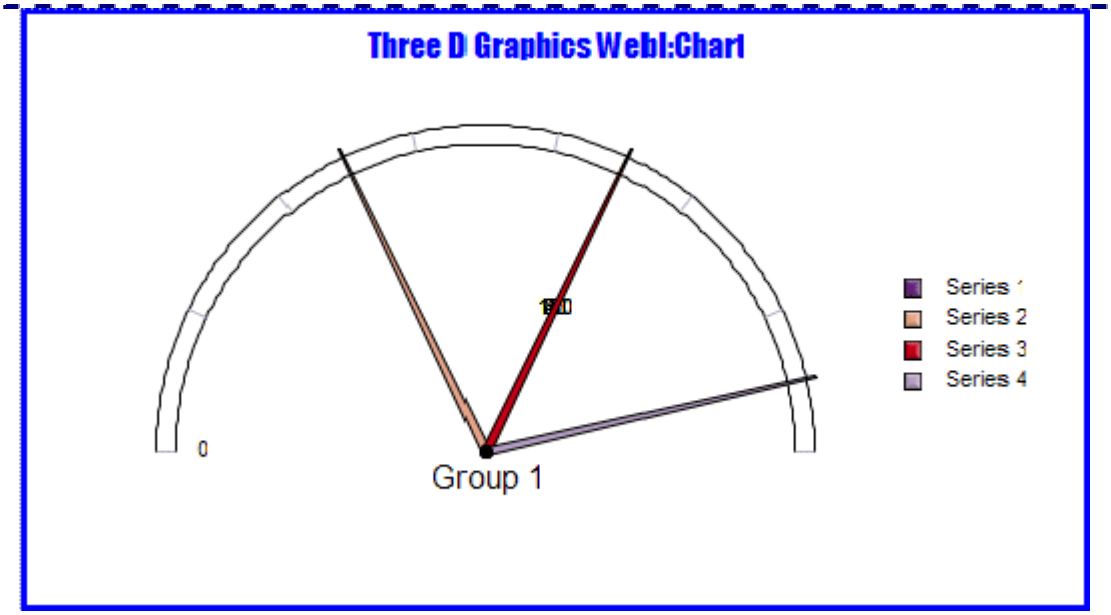
bEnable; 0=Use standard gauge drawing engine. 1=Use Enhanced Gauge Drawing Engine.

EXAMPLE:

```
@GAUGE_STYLE 1
```




```
@GAUGE_STYLE 0
```



PERSISTENT:

NO

@GAUGE_THRESHOLD (Gauge Band Threshold)

This macro changes the threshold value between the color bands in a gauge chart.

SYNTAX:

```
@GAUGE_THRESHOLD nBand fThreshold
```

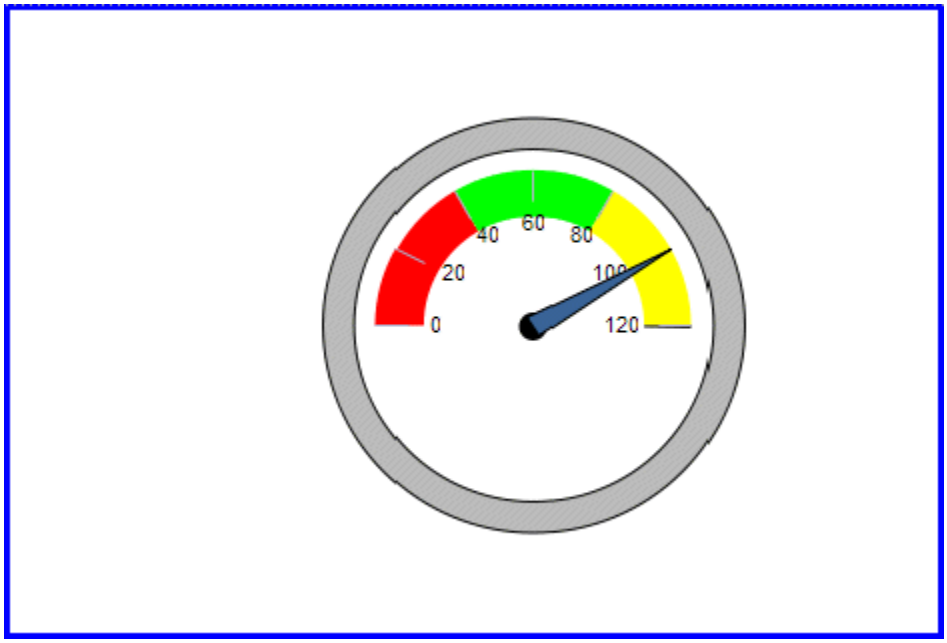
PARAMETERS:

nBand; 0...4 selects the color band

fThreshold; selects the threshold between color band *nBand* and *nBand*+1

EXAMPLE:

```
@GAUGE_THRESHOLD 0 40
@GAUGE_COLOR 0 255 0 0
@GAUGE_THRESHOLD 1 80
@GAUGE_COLOR 1 0 255 0
@GAUGE_THRESHOLD 2 120
@GAUGE_COLOR 2 255 255 0
```



PERSISTENT:

NO


WebI: CHART

Section 14: Pie Chart Macros

These macros can be used to format and control the appearance of pie charts:

- @DATATEXT_PIE; Control the appearance of data text in a pie chart
- @EXPLODE_SLICE; Explode slice from pie
- @PIE_DEPTH; Pie Depth
- @PIE_HOLE; Size of center ring in donut pie chart
- @PIE_ROTATE; Set Pie Rotation Start Point
- @SMART_PIE_LABELS; Enable/Disable Enhanced pie label layout engine
- @SMART_PIE_SETTINGS; Customize Enhanced pie label layout engine

@DATATEXT_PIE (Data Text Mode for Pie Charts)

This macro sets the data text mode in pie charts.

SYNTAX:

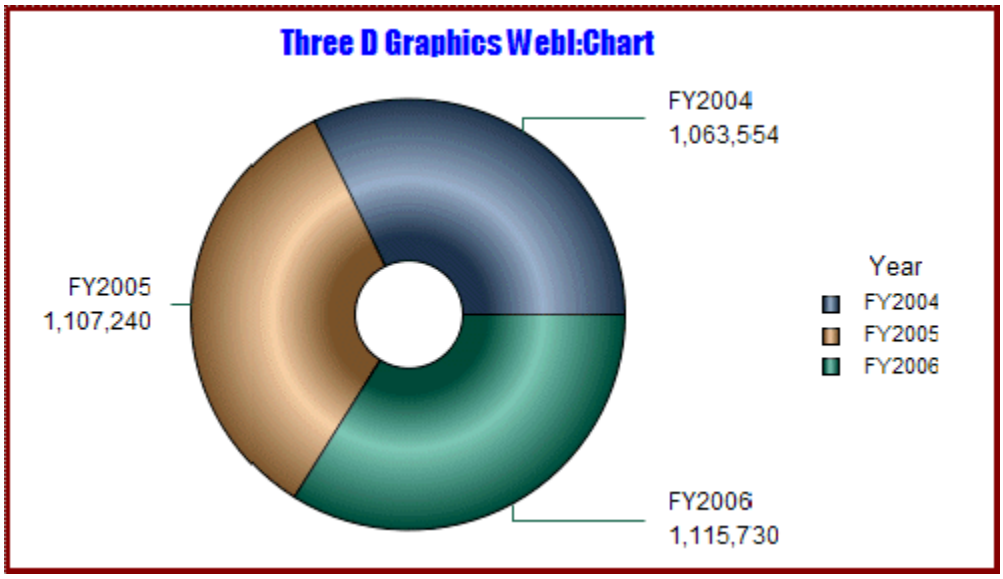
```
@DATATEXT_PIE nValue
```

PARAMETERS:

nValue; Selects the data text to show. It can be one of the following:
0 = NO Labels on Pie
1 = VALUE labels on Pie
2 = Series Labels on Pie
3 = Value and Series Labels on Pie

EXAMPLE:

```
@DATATEXT_PIE 3
```



PERSISTENT:

NO

@EXPLODE_SLICE (Explode Slice from Pie)

This macro explodes a slice from a pie chart.

SYNTAX:

```
@EXPLODE_SLICE nSeries nMove
```

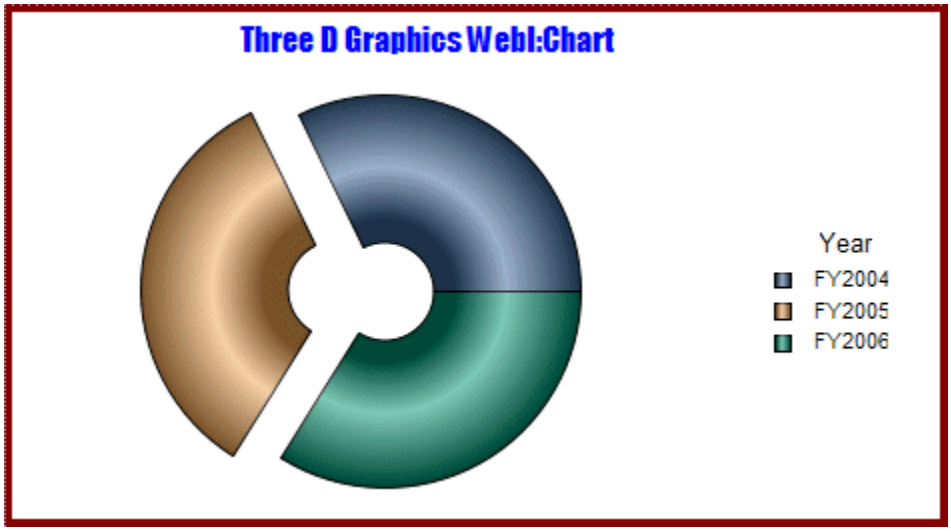
PARAMETERS:

nSeries; Series #

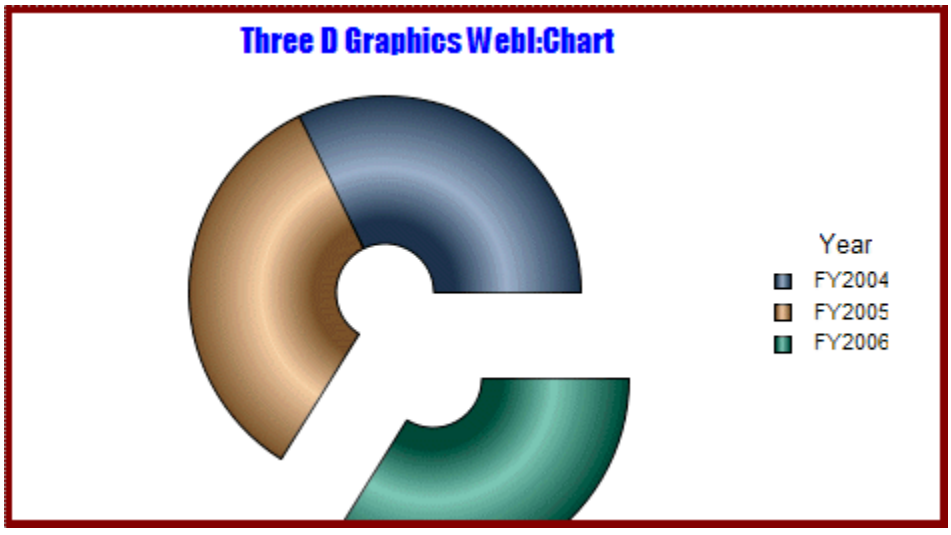
nMove; 0...100

EXAMPLE:

```
@EXPLODE_SLICE 1 25
```




```
@EXPLODE_SLICE 2 50
```



@PIE_DEPTH (Pie Depth)

The macro defines the depth of a pie chart.

SYNTAX:

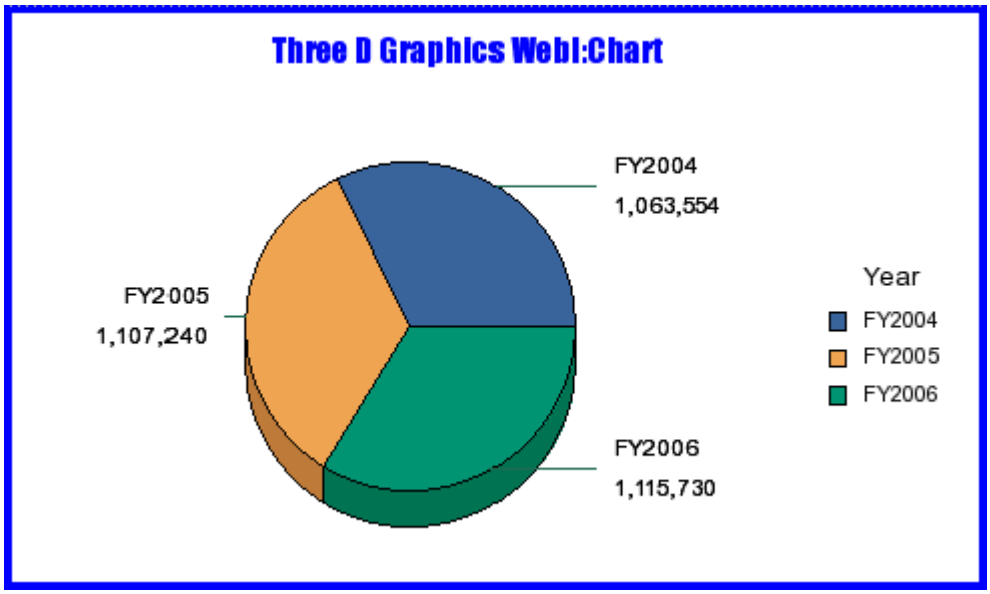
```
@PIE_DEPTH nPieDepth
```

PARAMETERS:

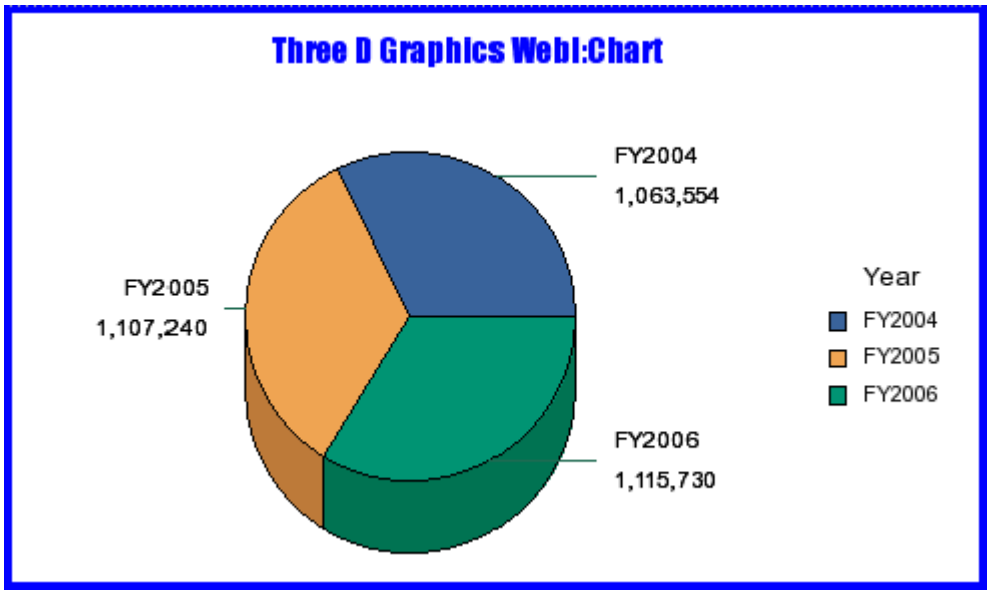
nPieDepth; 0...2000

EXAMPLE:

```
@PIE_DEPTH 500
```




```
@PIE_DEPTH 1000
```



@PIE_HOLE (Pie Hole)

This macro defines the size of the center ring in donut pie charts

SYNTAX:

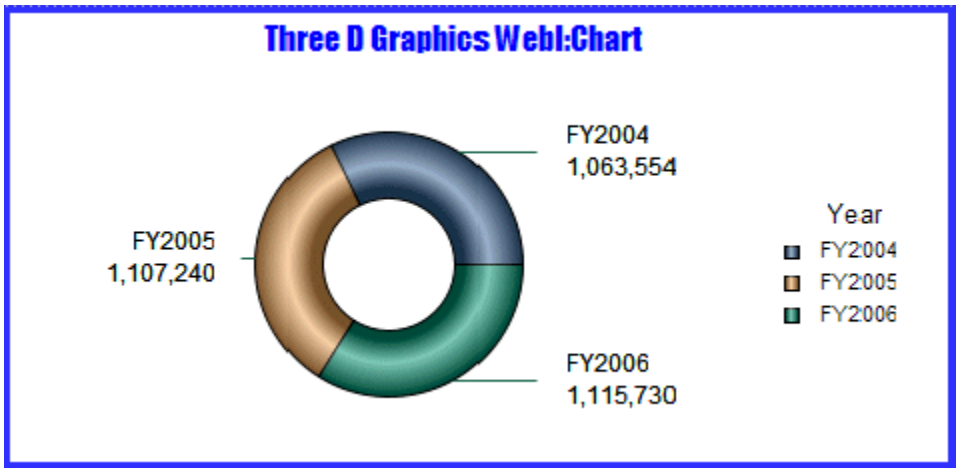
```
@PIE_HOLE nHoleSize
```

PARAMETERS:

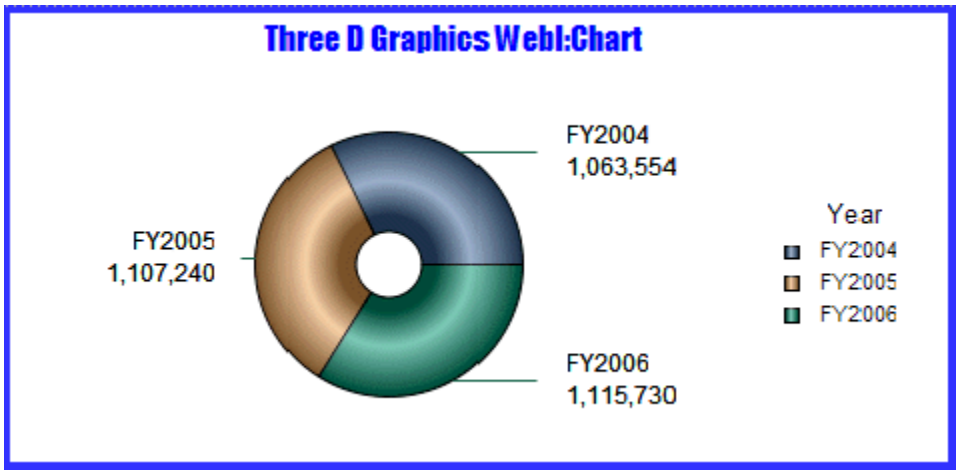
nHoleSize; 0...100

EXAMPLE:

```
@PIE_HOLE 50
```




```
@PIE_HOLE 25
```



@PIE_ROTATE (Pie Rotation Start Point)

This macro specifies the rotation start angle for pie charts.

SYNTAX:

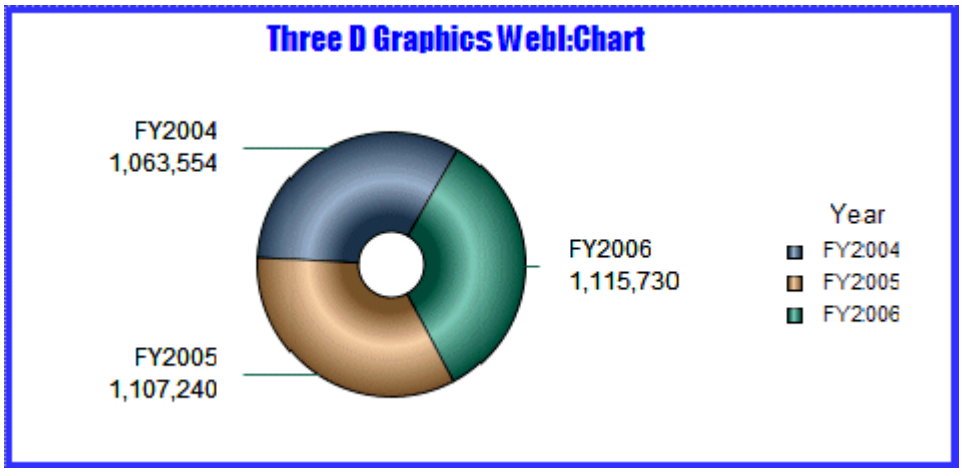
```
@PIE_ROTATE nRotation
```

PARAMETERS:

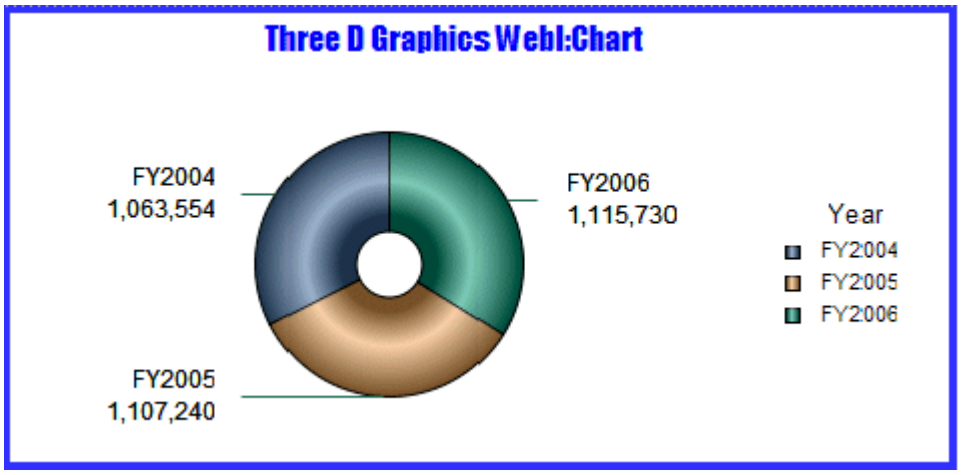
nRotation; 0...359 degrees to rotate pie

EXAMPLE:

```
@PIE_ROTATE 60
```




```
@PIE_ROTATE 90
```



PERSISTENT:

NO

@SMART_PIE_LABELS (Smart Pie Labels)

This macro enables/disables the enhanced pie label layout engine.

SYNTAX:

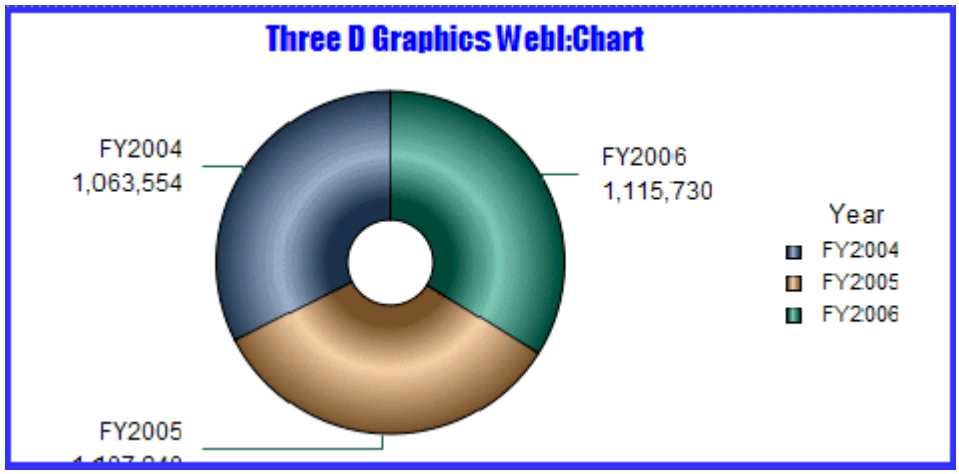
```
@SMART_PIE_LABELS bEnable
```

PARAMETERS:

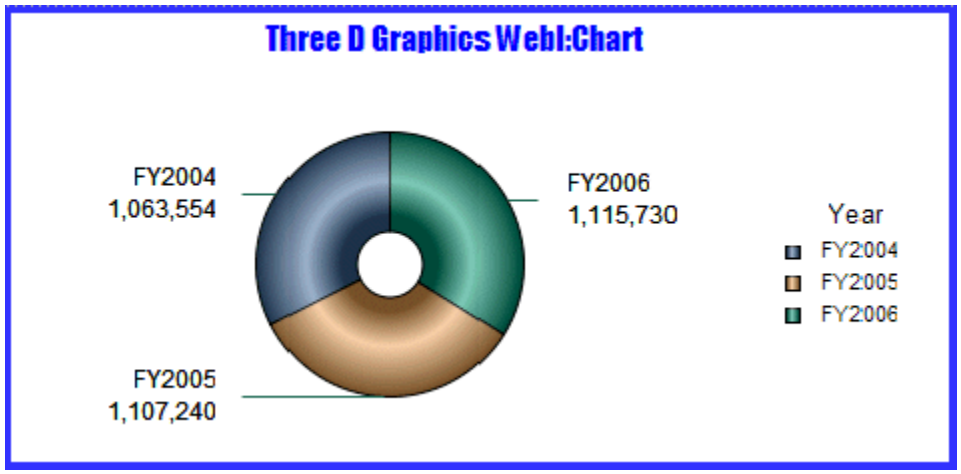
bEnable; 0=use standard pie engine, 1=use enhanced pie label layout engine

EXAMPLE:

```
@SMART_PIE_LABELS 0
```




```
@SMART_PIE_LABELS 1
```



PERSISTENT:

NO

@SMART_PIE_SETTINGS (Customize Smart Pie Labels)

When the enhanced pie label layout engine is enabled with @SMART_PIE_LABELS 1, this macro can be used to fine-tune the treatment of pie labels.

SYNTAX:

```
@SMART_PIE_SETTINGS nShrinkLabel nTruncateLabel nMoveLabel bShrinkPie
nPadLabel bReorderSlice bLabelInsidePie
```

PARAMETERS:

nShrinkLabel; -1...8
-1 = Use the Pie Label Placement Default Setting
0 = DO NOT shrink pie label font sizes
1...8 = Number of font sizes to evaluate to achieve the best result.

nTruncateLabel; -1...*nShrinkLabel*+1
-1 = Use the Pie Label Placement Default Setting
1...*nShrinkLabel*+1 = Start truncating labels after this many font sizes have been evaluated. Use a value of *nShrinkLabel*+1 to ensure that labels are NOT truncated.

nMoveLabel; -1...5
-1 = Use the Pie Label Placement Default Setting
0...5 = How aggressive to be when moving labels

bShrinkPie; -1...0
-1 = Use the Pie Label Placement Default Setting
0 = DO NOT Shrink pie radius before shrinking fonts
1 = Shrink pie radius before shrinking fonts

nPadLabel;
-1 = Use the Pie Label Placement Default Setting
0...5 = Amount of label padding

bReorderSlice; -1...0
-1 = Use the Pie Label Placement Default Setting
0 = DO NOT Reorder slices
1 = Reorder slices as needed

bLabelInsidePie; -1...0
-1 = Use the Pie Label Placement Default Setting
0 = DO NOT Force labels inside or outside the pie
1 = Force some labels inside or outside the pie as needed

PERSISTENT:

NO

Section 15: Macros for Waterfall Charts

These macros create and format waterfall charts:

- @WATERFALL; Create a normal Waterfall Chart
- @WATERFALL2; Create a Waterfall Chart with Total Group
- @WF_CONNECT; Assign a line style to feeler lines in a waterfall chart

@WATERFALL (Waterfall Chart)

This macro creates a waterfall chart.

SYNTAX:

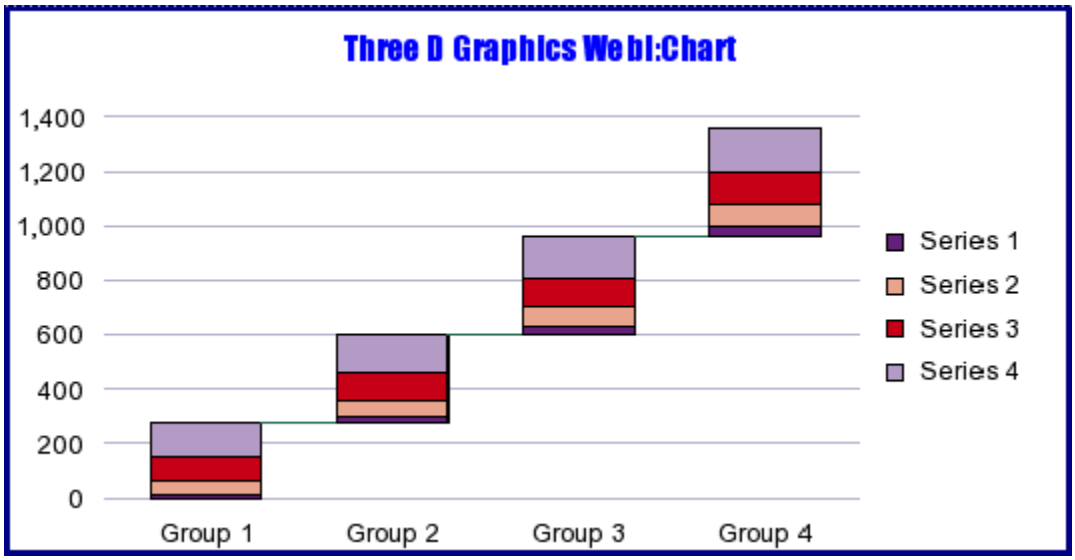
```
@WATERFALL
```

PARAMETERS:

None

EXAMPLE:

```
@WATERFALL
```



PERSISTENT:

YES

ALSO SEE:

@WATERFALL2

@WATERFALL2 (Waterfall Chart with Total Group)

This macro creates a waterfall chart and forces the last group in the chart to be a TOTAL. This forces the last value to start at the base line instead of being another stair in the waterfall's staircase.

SYNTAX:

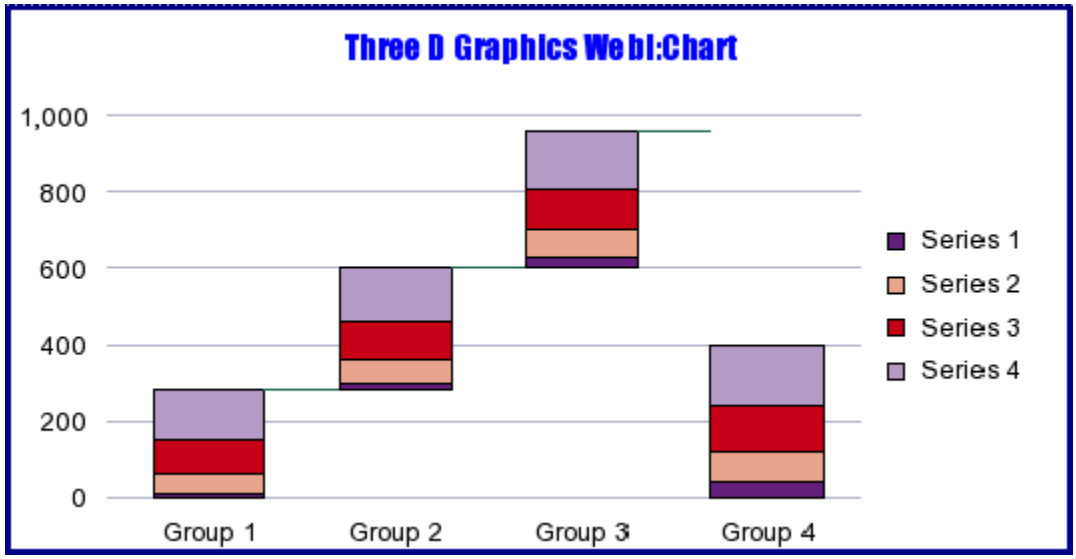
```
@WATERFALL2
```

PARAMETERS:

None

EXAMPLE:

```
@WATERFALL2
```



PERSISTENT:

YES

ALSO SEE:

@WATERFALL

@WF_CONNECT (Waterfall Feeler Lines)

This macro assigns a line style to feeler lines in a waterfall chart.

SYNTAX:

```
@WF_CONNECT nStyle
```

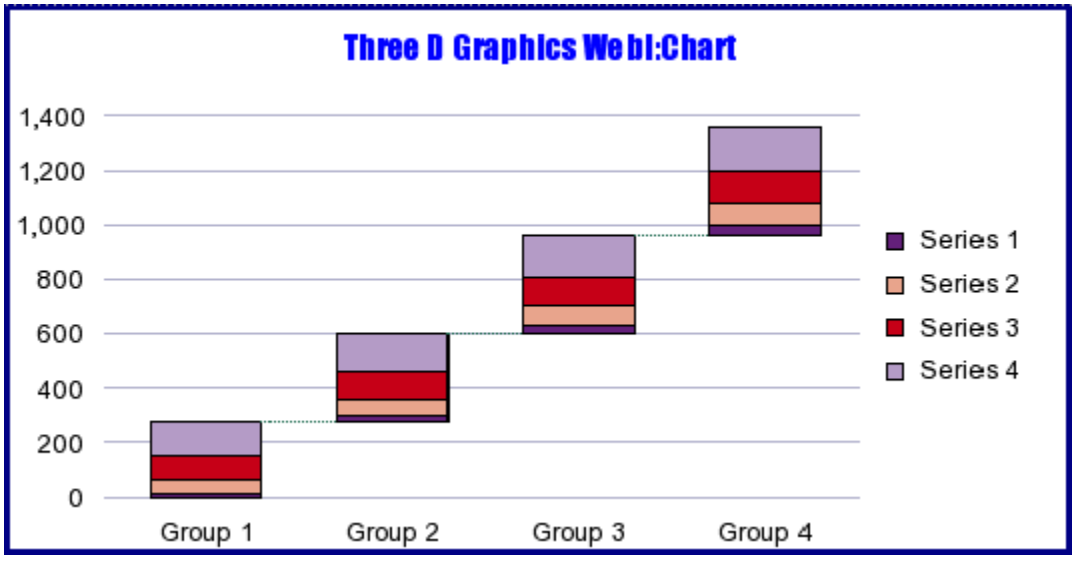
PARAMETERS:

nStyle; 0...7 selects one of the following line styles.

0 = Solid
1 = Dashed
2 = Dotted
3 = Dot-Dash
4 = Dash-Dot-Dot
5 = Medium Dash
6 = Short Dash
7 = Long Dash

EXAMPLE:

```
@WATERFALL2
@WF_CONNECT 2
```



PERSISTENT:

YES

Section 16: Miscellaneous

- @EB; Adds error bars to standard bar/column charts
- @FONTANGLE; Change the rotation angle of a text object
- @FONTNAME; Map a global font into particular font object on the chart
- @FONTSIZE; Change the font size of a text object
- @FRAME; Define frame size/location
- @HAT; Define the size of hats on Error Bars
- @LEGEND; Define legend size/location
- @LEGEND_ORDER; Force Legend Order
- @LEGEND_WRAP_WIDTH; Define the virtual wrap point for all legend text
- @ORD_SPACE; Extend Line/Area Charts to Frame Edge

@EB (Error Bars)

This macro adds error bars to standard bar/column charts. Your data must be in the following form:

Value 1: Series Value

Value 2: Error High Value

Value 3: Error Low Value

SYNTAX:

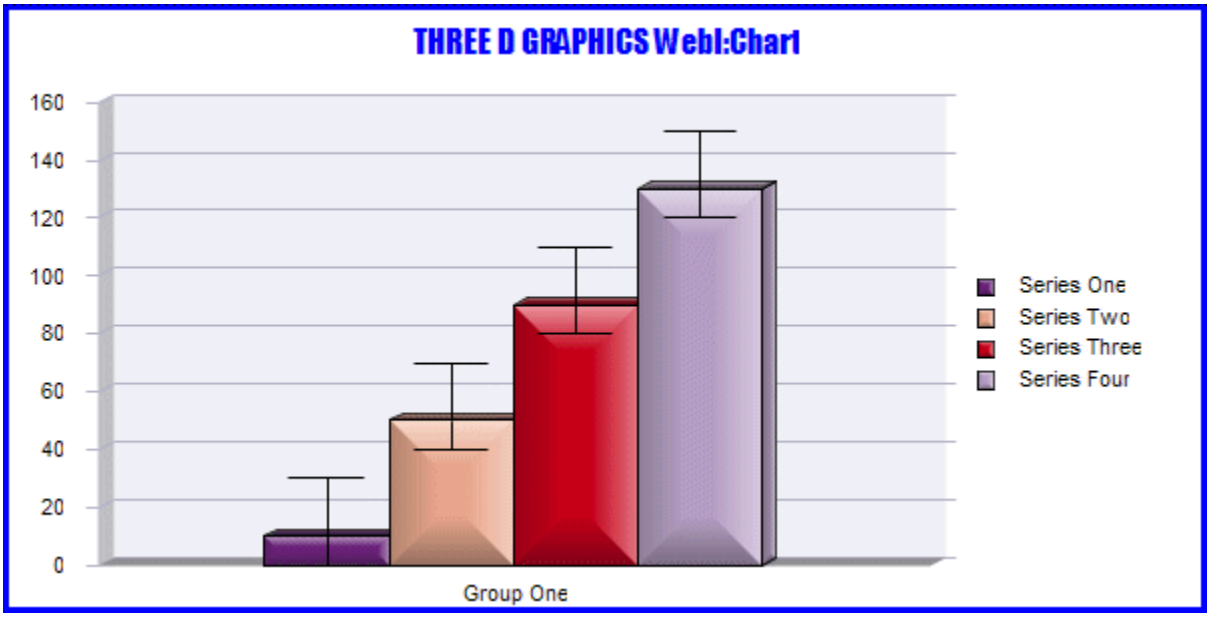
```
@EB bShow
```

PARAMETERS:

bShow; 0 = turn off error bars, 1 = turn on error bars

SYNTAX:

```
@EB 1
```



PERSISTENT:

NO

ALSO SEE:

@HAT (Hat on Error Bars)

@FONTANGLE (Font Angle)

This macro can be used to change the rotation angle of a text object in a chart.

SYNTAX:

```
@FONTANGLE nObject nAngle
```

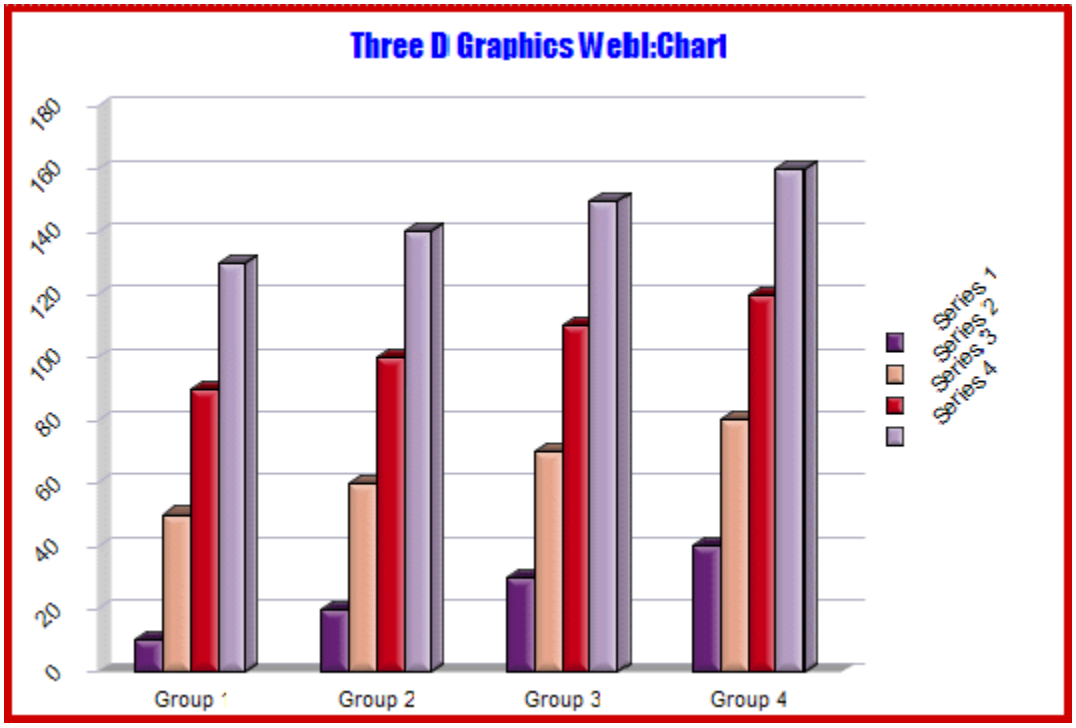
PARAMETERS:

nObject; 1...11 selects one of the following chart objects:
1 = Title
3 = X1-Axis Title
4 = Y1-Axis Title
5 = Y2-Axis Title
7 = Data Text
8 = O1-Axis Labels or X1-Axis Labels depending on the chart type
9 = Y1-Axis Labels
10 = Y2-Axis Labels
11 = Legend Text

nAngle; 0...6 applies one of the following angles to *nObject*:
0 = Normal, Horizontal Characters
2 = Rotate Characters 90 Degrees
3 = Rotate Characters 180 Degrees
4 = Rotate Characters 270 Degrees
5 = Rotate Characters 45 Degrees
6 = Rotate Characters 315 Degrees

EXAMPLE:

```
@FONTANGLE 9 5 @FONTANGLE 11 5
```



PERSISTENT:

YES

@FONTNAME (Font Name)

This macro can be used to assign a global font to a font object in a chart. It is useful when you need to dynamically change a font to look better in another language (Japanese, for example).

SYNTAX:

```
@FONTNAME nObject nFontIndex
```

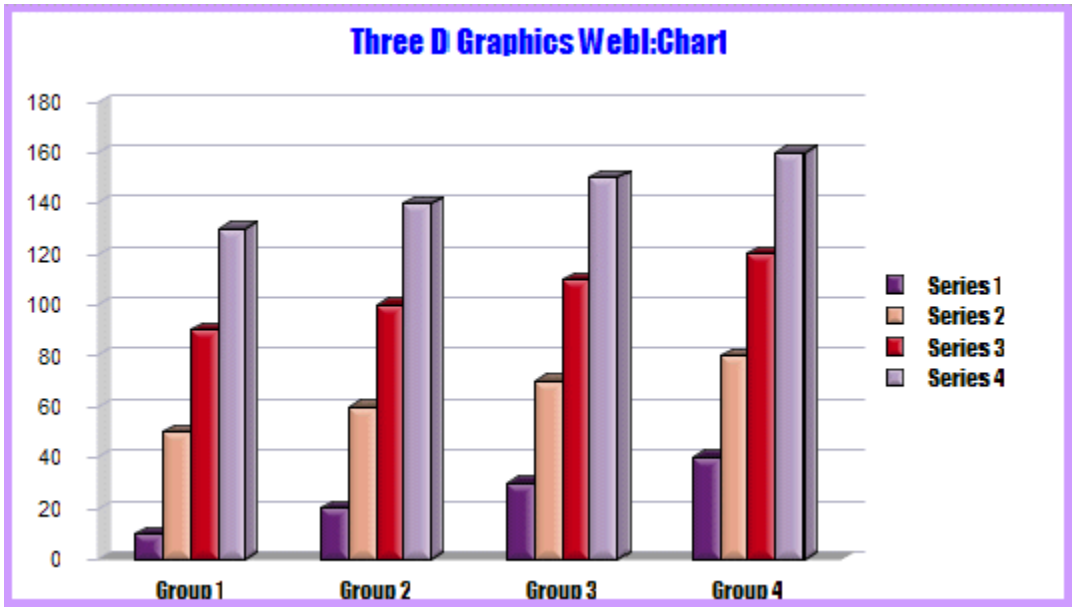
PARAMETERS:

nObject; 1...11 selects one of the following chart objects:
1 = Title
3 = X1-Axis Title
4 = Y1-Axis Title
5 = Y2-Axis Title
7 = Data Text
8 = O1-Axis Labels or X1-Axis Labels depending on the chart type
9 = Y1-Axis Labels
10 = Y2-Axis Labels
11 = Legend Text

nFontIndex; 0...2 selects one of the following global fonts
0=Arial
1=MS PGothic
2=Impact

EXAMPLE:

```
Three D Graphics WebI:Chart~ @FONTNAME 1 2 @FONTNAME 8 2
@FONTNAME 11 2
```



PERSISTENT:

YES

@FONTSIZE (Font Size)

This macro sets the point size of a text object in a chart.

SYNTAX:

```
@FONTSIZE nObject nSize
```

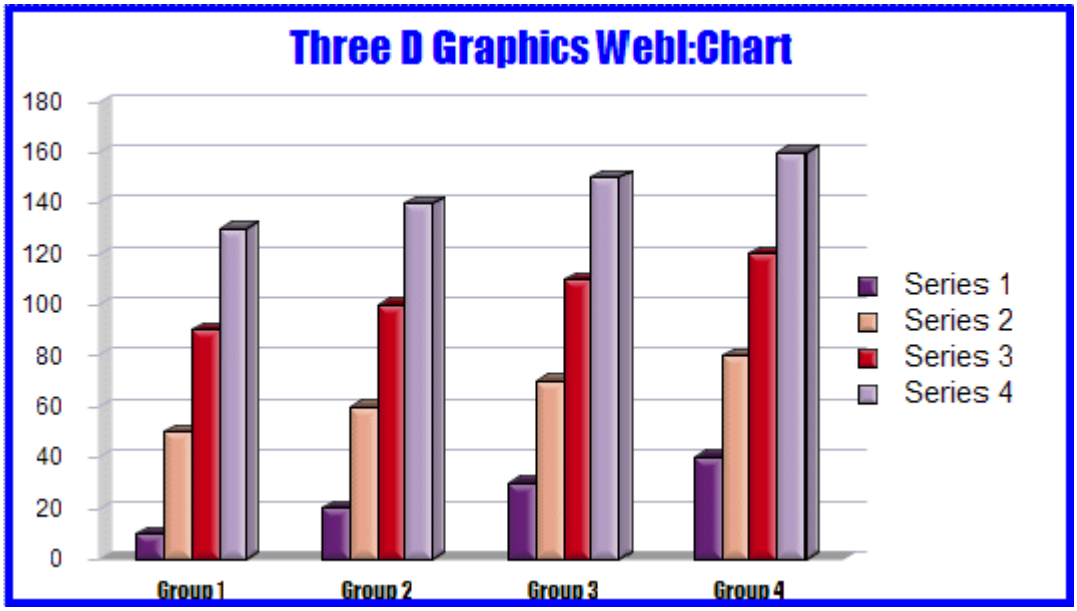
PARAMETERS:

nObject; 1...11 selects one of the following chart objects:
1 = Title
3 = X1-Axis Title
4 = Y1-Axis Title
5 = Y2-Axis Title
7 = Data Text
8 = O1-Axis Labels or X1-Axis Labels depending on the chart type
9 = Y1-Axis Labels
10 = Y2-Axis Labels
11 = Legend Text
12 = Series labels in Pie Charts
13 = Value labels in Pie Charts

nSize; any INT16 value that defines a font point size

EXAMPLE:

```
Three D Graphics WebI:Chart~ @FONTNAME 1 2 @FONTSIZE 1 16
@FONTNAME 8 2 @FONTSIZE 11 12
```



PERSISTENT:

NO

@FRAME (Frame Size/Location)

This macro can be used to specify a fixed size/location for the chart frame. If you want the chart frame to be the exact same size and in the same location on each page of a report, this macro will position the chart frame at the specified X/Y coordinates.

SYNTAX:

```
@FRAME nUpperLeftX nUpperLeftY nLowerRightX nLowerRightY
```

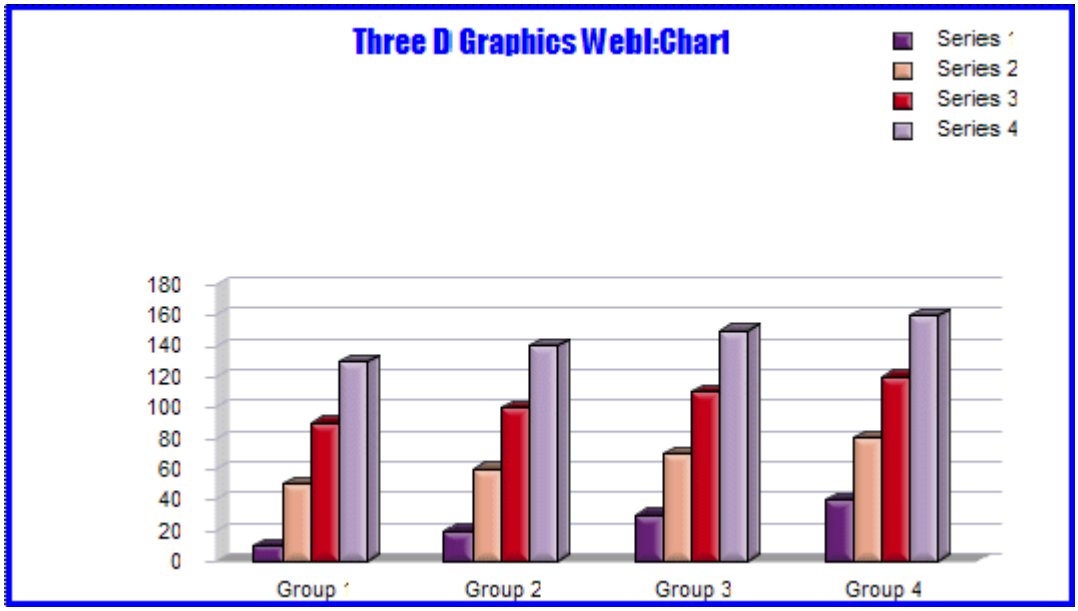
PARAMETERS:

nUpperLeftX, nUpperLeftY; -16382...+16382 specifies the X/Y position of the upper left corner of the chart frame in virtual coordinates

nLowerRightX, nLowerRightY; -16382...+16382 specifies the X/Y position of the lower right corner of the chart frame in virtual coordinates

EXAMPLE:

```
@FRAME -10000 1110 13383 -14000
@LEGEND 16000 16000 16300 16300
```



PERSISTENT:

NO

@HAT (Hat on Error Bars)

This macro sets the width of the “hat” portion of an error bar that is created by the @EB macro.

SYNTAX:

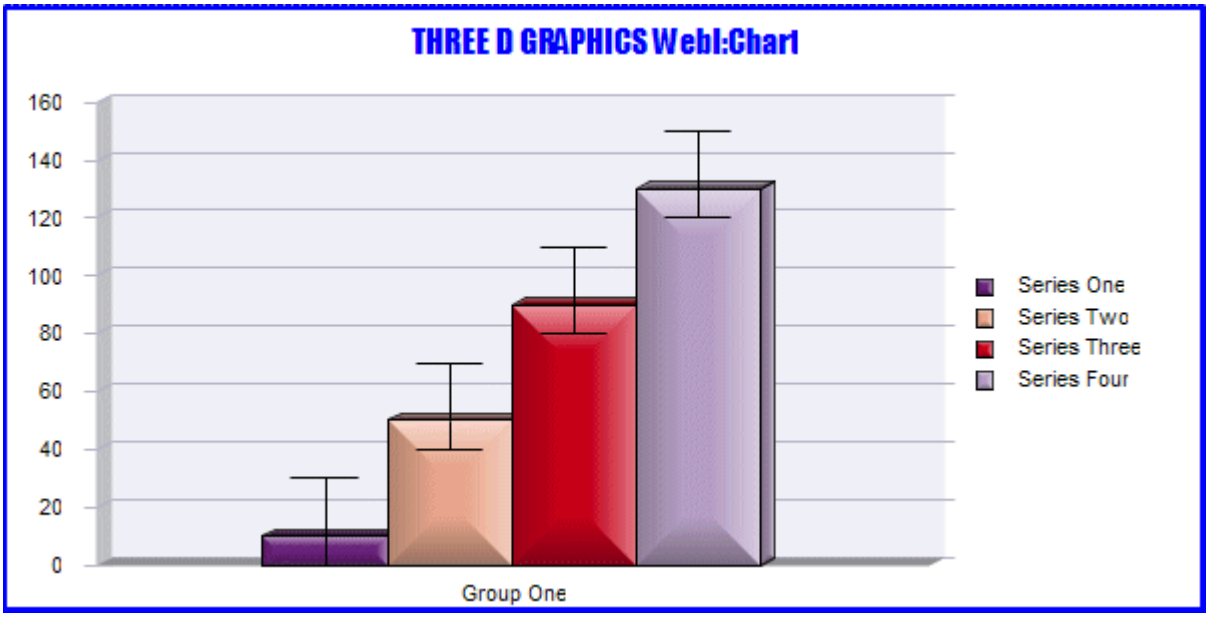
```
@HAT nWidth
```

PARAMETERS:

nWidth; Width of Hat (0...100), 0=No Hat, 100=Widest Possible Hat.

EXAMPLE:

```
@HAT 90
```



PERSISTENT:

NO

ALSO SEE:

@EB

@LEGEND (Legend Size/Location)

This macro can be used to specify a fixed size/location for the chart legend. If you want the chart legend to be the exact same size and in the same location on each page of a report, this macro will position the chart legend at the specified X/Y coordinates.

SYNTAX:

```
@LEGEND nUpperLeftX nUpperLeftY nLowerRightX nLowerRightY
```

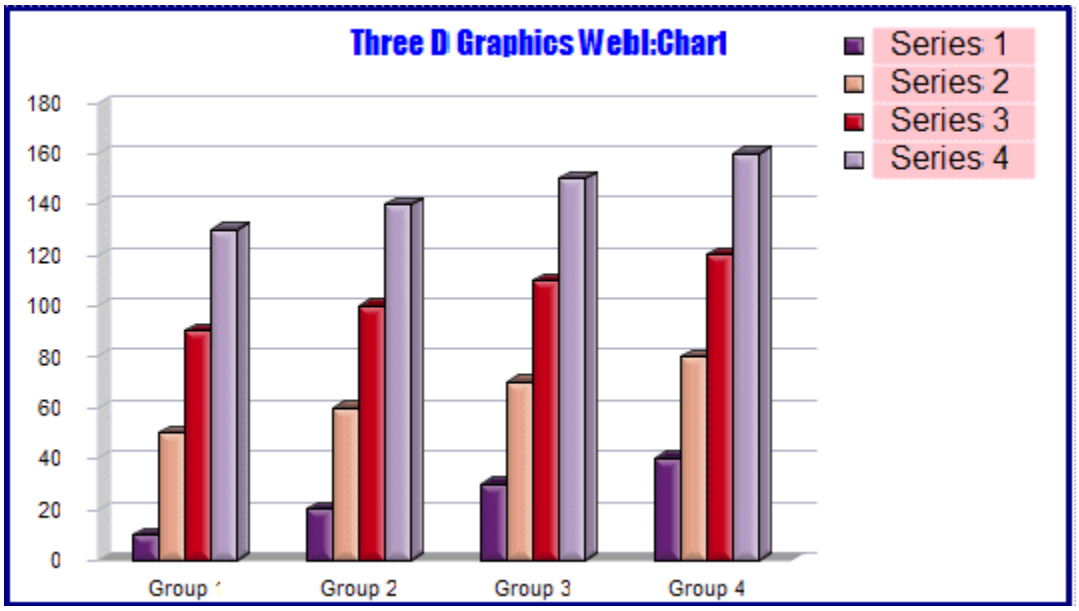
PARAMETERS:

nUpperLeftX, nUpperLeftY; -16382...+16382 specifies the X/Y position of the upper left corner of the chart legend in virtual coordinates

nLowerRightX, nLowerRightY; -16382...+16382 specifies the X/Y position of the lower right corner of the chart legend in virtual coordinates

EXAMPLE:

```
@LEGEND 16000 16000 16300 16300 @GCOLOR 11 Pink
```



PERSISTENT:

NO

@LEGEND_ORDER (Force Legend Order)

This macro can be used to control the order in which series are drawn in the legend.

SYNTAX:

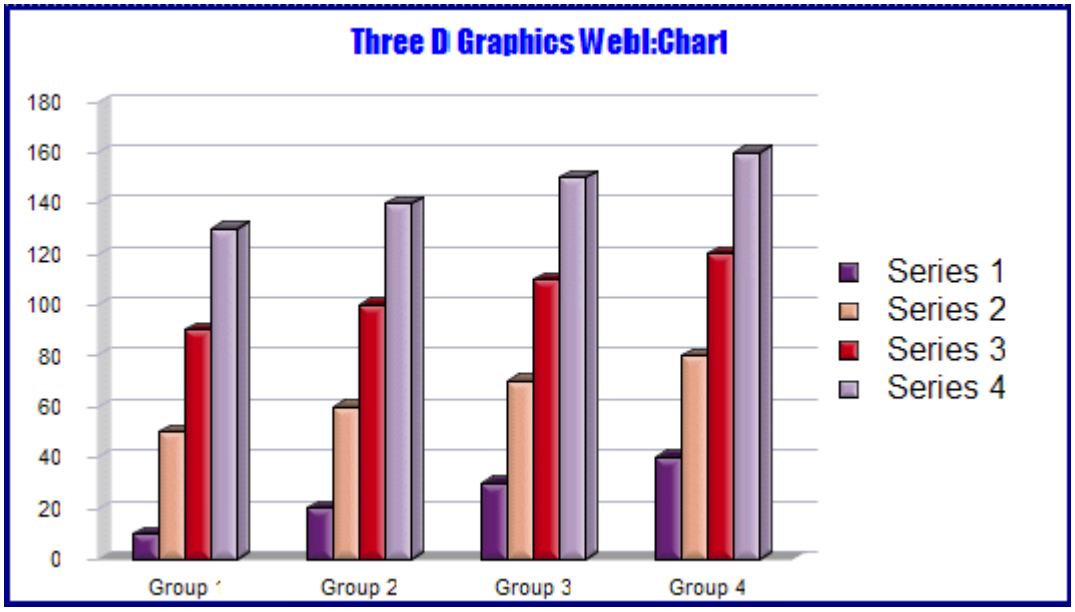
```
@LEGEND_ORDER nForceLegendOrder
```

PARAMETERS:

nForceLegendOrder; 0...2.
0 = Use internal logic to determine legend order
1 = Force legend order to Series 0.... Series N
2 = Force legend order to Series N .... Series 0

EXAMPLE:

```
@LEGEND_ORDER 1
```



PERSISTENT

NO

@LEGEND_WRAP_WIDTH (Legend Wrap Width)

This macro can be used to define a virtual wrap point for all legend text. It overrides the system default of 6500 to create very wide legends that do not wrap onto a second line.

SYNTAX:

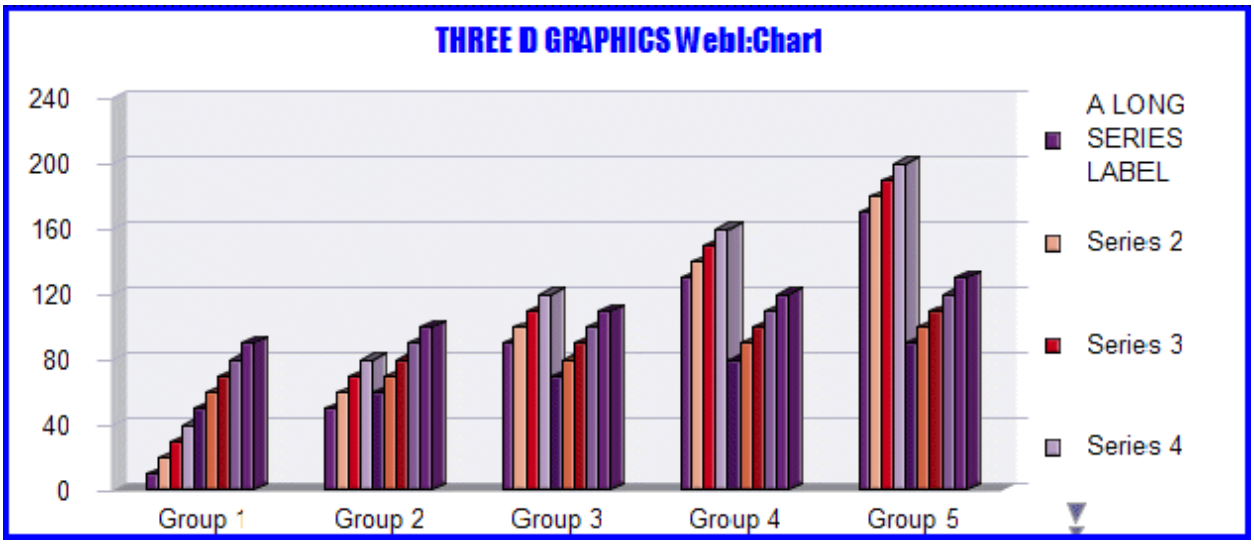
```
@LEGEND_WRAP_WIDTH nWrapPoint
```

PARAMETERS:

nWrapPoint; 0...16000 defines the virtual wrap point for all legend text

EXAMPLE:

```
@ASL 2 A LONG SERIES LABEL
@LEGEND_WRAP_WIDTH 1200
```



PERSISTENT:

NO

@ORD_SPACE (Extend Line/Area Charts to Frame Edge)

This macro can be used to specify how line and area charts are drawn. Set *nMode* to zero to select normal drawing mode - inset from the chart frame. Set *nMode* to one to select extended drawing mode - chart is extended to chart frame.

SYNTAX:

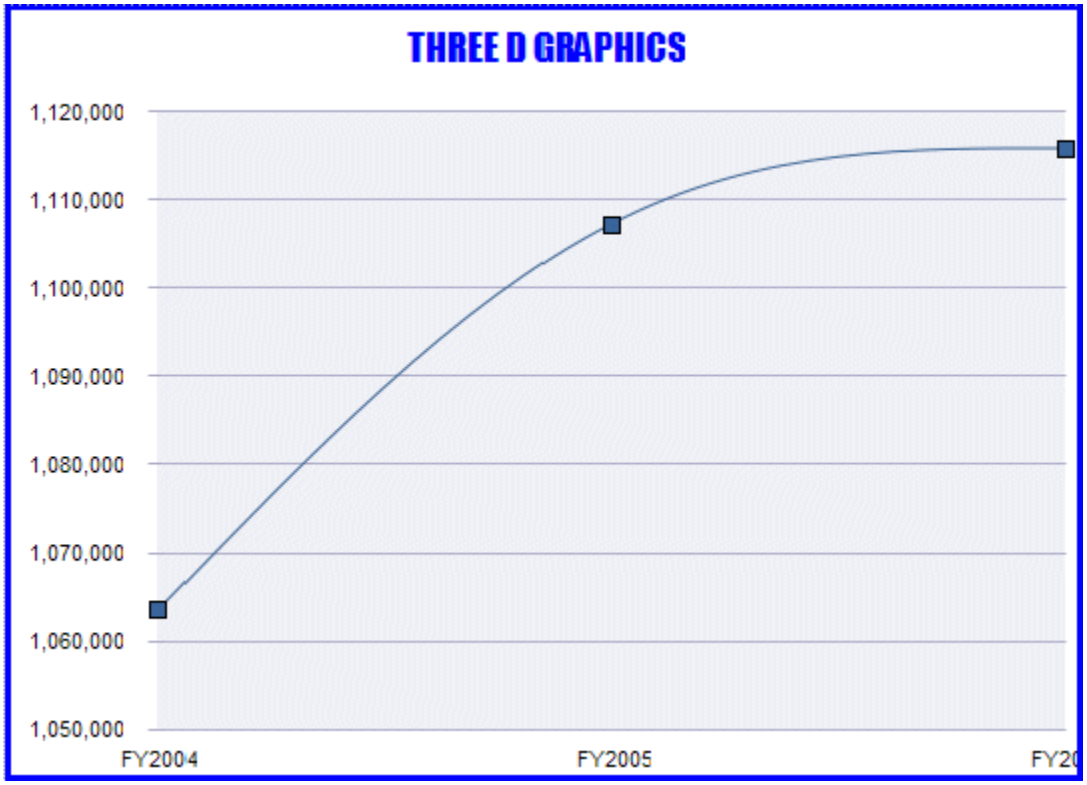
```
@ORD_SPACE nMode
```

PARAMETERS:

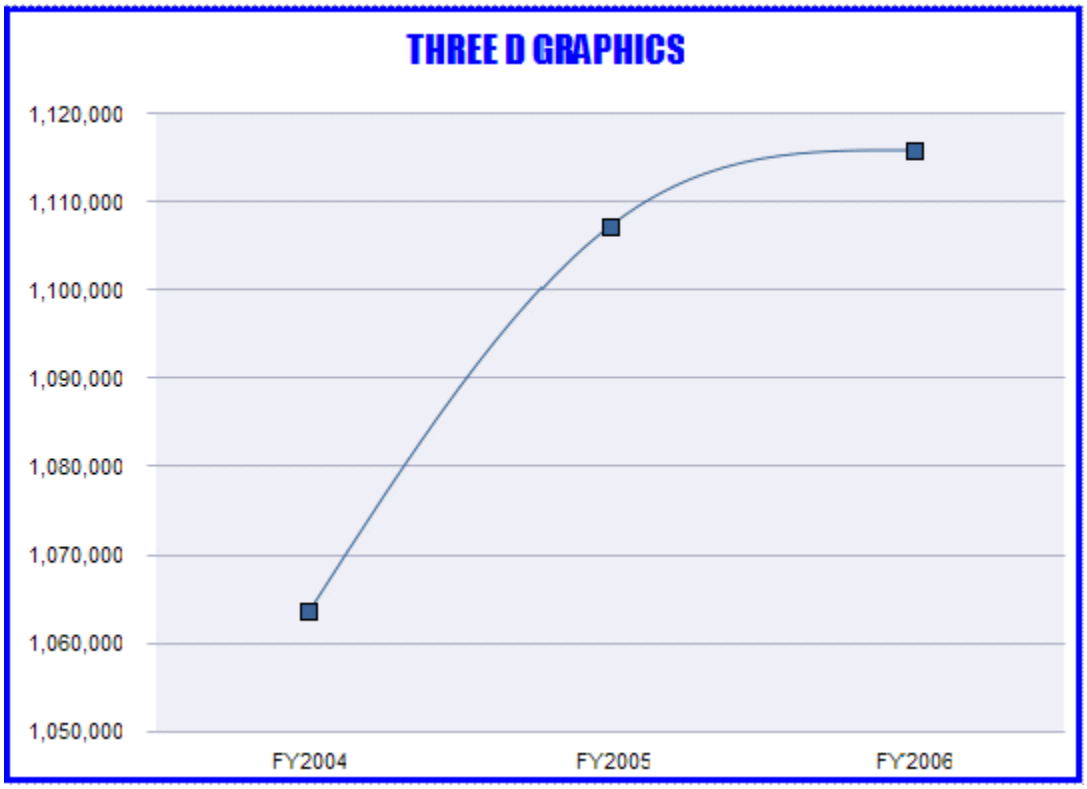
nMode; 0=normal drawing mode (inset from chart frame), 1=extended drawing mode (draw chart extended to frame edge)

EXAMPLE:

```
@ORD_SPACE 1
```




```
@ORD_SPACE 0
```



PERSISTENT:

NO

Section 17: Troubleshooting

- @DEBUG; Show Debug Information
- @PARAM_COUNT_FIXUP; Debug Macro
- @PARAM_FIXUP; Fix parameters from Fields/Functions.
- @RESET; Reset Internal Data Range

@DEBUG (Show Debug Information)

This macro provides useful information for tracking problems that may occur in WebI:Chart. Do not use this macro unless you are instructed to do so by Three D Graphics technical support.

SYNTAX:

```
@DEBUG
```

PARAMETERS:

None

PERSISTENT:

NO

@PARAM_COUNT_FIXUP (Debug Macro)

This macro provides useful information for tracking problems that may occur in WebI:Chart. Do not use this macro unless you are instructed to do so by Three D Graphics technical support.

SYNTAX:

```
@PARAM_COUNT_FIXUP nFixup
```

PARAMETERS:

nFixup; -1...9

PERSISTENT:

N/A

@PARAM_FIXUP (Parameter Fix-Up)

This macro provides useful information for tracking problems that may occur in WebI:Chart. Do not use this macro unless you are instructed to do so by Three D Graphics technical support.

SYNTAX:

```
@PARAM_FIXUP nIndex
```

PARAMETERS:

nIndex; 1=Enable parameter fix-up, 0=Disable parameter fix-up

PERSISTENT:

NO

@RESET (Reset Internal Data Range)

In some unusual cases, WebI:Chart macros are not applied until the user clicks on the chart. This macro solves this problem. It will reset the internal data range (useful for tracking parameter substitution errors). Do not use this macro unless you are instructed to do so by Three D Graphics technical support.

SYNTAX:

```
@RESET
```

PARAMETERS:

None

PERSISTENT:

NO


WebI: CHART