THREE D GRAPHICS

(Chart Enhancement Macros for Cognos Series 7)

Version: Beta

MARCH, 2009

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

Copyright:

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

MARCH, 2009

Warranty:

Three D Graphics is solely responsible for the development, product testing, system testing, production, marketing and support of the current and future versions of ***Charts Unlimited*** for IBM Cognos Series 7/Impromptu. Users can be assured that the IBM Cognos-provided chart component of Series 7/Impromptu can be single-click restored at any time. In such event, Three D Graphics will cheerfully refund the unused portion of any of the selected ***Charts Unlimited*** licensing options.

As noted therein, Three D Graphics is the Licensor to IBM Cognos of the current Impromptu Series 7 chart technology, as well as the chart component in IBM Cognos 8 BI. Accordingly, the Company is in the best position to offer the comprehensive benefits of ***Charts Unlimited*** to those Users of Series 7/Impromptu that have not yet elected to migrate to the overall substantial attributes of IBM Cognos 8 BI.

Table of Contents

Section 1: Overview 1-1
System Requirements 1-1
Installation & Setup 1-1
Getting Started 1-3
Order of Evaluation 1-4
Persistence 1-5
Error Reporting 1-6
Using Fields/Functions in Macros 1-6
Section 2: Auto Arrange Macros 2-1
@AA (Auto Arrange) 2-2
@AA2 (Auto Arrange with Frame Adjustment) 2-3
@AA3 (Auto Arrange with Frame & Legend Adjustment) 2-5
@AA4 (Auto Arrange with Maximized Frame Adjustment) 2-7
Section 3: Axis Macros 3-1
@AXIS (Assign Series to Axis) 3-2
@DX (X-Axis Divisions) 3-3
@DY (Y-Axis Divisions) 3-4
@DY2 (Y2-Axis Divisions) 3-5
@GX (X-Axis Grid Style) 3-6
@GY (Y-Axis Grid Style) 3-7
@HOUR_SCALE (Minor Axis Hour Scale) 3-8
@PX (X-Axis Precision) 3-9
@PY (Y1-Axis Precision) 3-10
@PY2 (Y2-Axis Precision) 3-11
@SC (Y-Axis Scale) 3-12
@SCX (X-Axis Scale) 3-13
@SCY2 (Y2-Axis Scale) 3-14
@X_AXIS_MODE (X-Axis Mode) 3-15
@Y1_FORCE_PERCENT (Y1-Axis Percent Format) 3-16
@Y1_FORCE_PERCENT2 (Y1-Axis Percent/Numeric Format) 3-17
@Y1_INVERT (Y1-Axis Invert) 3-18
@Y1BASE (Y1-Axis Base Line) 3-19
@Y2_FORCE_PERCENT (Y2-Axis Percent Format) 3-20
@Y2_INVERT (Y2-Axis Invert) 3-21
@Y2BASE (Y2-Axis Base Line) 3-22
@Y2SLAVE (Y2-Axis Slave to Y1) 3-23
@Y2SLAVE2 (Y1/Y2 Slave to Max Value) 3-24
Section 4: Series & Groups 4-1
@CALC_PERCENT_SERIES (Calculate Percent Series) 4-2
@FORCE_ABSOLUTE (Force Series Absolute) 4-3
@FORCE_SERIES_COUNT (Force Series Count) 4-4
@FORECAST (Add Blank Groups) 4-5
@GM (Group Major) 4-6
@IG (Ignore Group) 4-7
@IS (Ignore Series) 4-8
@LIMIT_VISIBLE_GROUPS (Limit Visible Groups) 4-9
@MIN_GROUPS (Add Blank Groups/Control Bar Size) 4-10
@PERCENT_SERIES (Create Ratio Series) 4-11
@RG (Reverse Group) 4-12
@RS (Reverse Series) 4-13
@SINGLE_GROUP (Single Group in Scatter Chart) 4-14
@SORT (Sort Series/Groups) 4-15
@STEP_LINE (Series Stepped Line) 4-17

@STEP_LINE2 (Series Stepped Line at Values) 4-18
@SWAP (Swap Series/Groups) 4-19
@TOTAL_GROUP (Create a Total Group) 4-20
@USER_SERIES (User-Defined Series) 4-21
Section 5: Labels 5-1
@3DLABEL (Adjust Labels in a 3D Chart) 5-2
@AGL (Alias Group Label) 5-3
@ASL (Alias Series Label) 5-4
@ASL_DP (Alias Series Label/Data Point) 5-5
@COND_GROUP_LABEL (Conditional Group Label) 5-6
@COND_GROUP_LABEL2 (Conditional Group Label 2) 5-7
@GROUP_LABELS_ON_BASELINE (Group Labels on Baseline) 5-8
@XSKIP (X-Axis Skip Labels) 5-9
@XSKIP2 (X-Axis Skip/Force Last Label) 5-10
@XSKIP3 (Max X-Axis Labels/Auto Adjust Skip) 5-11
Section 6: Riser & Markers 6-1
@BUBBLEMODE (Bubble Size in Bubble Charts) 6-2
@HIDE_ZERO (Hide Zero Riser/Marker) 6-4
@HL (Highlight Riser/Marker) 6-5
@MARKER (Marker Shapes) 6-6
@MCOLOR (Marker Colors) 6-7
@PAT (Riser Pattern) 6-8
@RISER_BORDER (Riser Border) 6-9
@RISER_OVERLAP (Riser Overlap) 6-10
@RISER_WIDTH (Riser Width) 6-11
@SZ (Size of Markers) 6-12
Section 7: Data & Data Text 7-1
@APPEND_DATATEXT (Append String to Data Text) 7-2
@DATATEXT (Data Text Mode) 7-3
@DLT (Data Line Type) 7-4
@DP (Data Point Override) 7-5
@DPC (Data Point Clear) 7-6
@DT (Show/Hide Series-Specific Data Text) 7-7
@FORCE_DATALINE (Force Data Line) 7-8
@FORCE_DATATEXT_CURRENCY (Force Data Text to Currency Format) 7-9
@IR (Insert Row) 7-10
@OFFSCALE_Y1 (Y1-Axis Off-Scale Values) 7-11
@RDT (Rotate Data Text) 7-12
@Y_ZERO (Include/Exclude Zero in Y1-Auto-Scale) 7-13
Section 8: Color & Special Effects 8-1
@ALPHA (Alpha Channel Transparency) 8-2
@ALT_FRAME_FILL (Alternating Frame Fill) 8-3
@ALT_FRAME_FILL_MODE (Alternating Frame Fill Mode) 8-4
@AUTO_COLOR (Automatic Color Mode) 8-5
@BEVEL (Bevel Chart Object) 8-6
@COLOR_MODE (Color Mode) 8-7
@COLOR_SCHEME (Activate a Color Scheme) 8-8
@COND_COLOR & @COND_COLOR2 (Conditional Colors) 8-9
@CURVED_LINES (Enable Curved Lines) 8-11
@DEFINE_SCHEME (Define Color Scheme) 8-12
@GCOLOR (Color Chart Object) 8-13
@SHADOW (Drop Shadow) 8-14
Section 9: Macros for Lines 9-1
@CX (X-Axis Line with Color) 9-2

@CXY (X/Y Coordinates Line with Color) 9-3
@CY (Y1-Axis Line with Color) 9-4
@INIT_USERLINES (Initialize User-Defined Lines) 9-5
@LS (Line Style) 9-6
@MEAN (Mean Line) 9-7
@MOVA (Moving Average Line) 9-8
@SMOOTH_LINE (Smooth/Straight Lines) 9-9
@TRENDLINE (Trend Line) 9-10
@TRENDLINE_ALLDATA (Linear Regression Line) 9-11
@X (X-Axis Line at Value) 9-12
@XG (X-Axis Line at Group) 9-13
@XSZ (X-Axis Line with Label) 9-14
@XSZL (X-Axis Line with Label on Left/Lower) 9-15
@XSZN (X-Axis Line with Label & Value) 9-16
@XSZNL (X-Axis Line with Label & Value on Left/Lower) 9-17
@XY (X/Y Coordinates Line) 9-18
@XY_DP2 (Data Point Line) 9-19
@Y (Y1-Axis Line) 9-20
@YSZ (Y1-Axis Line with Label) 9-21
@YSZL (Y1-Axis Line with Label on Left) 9-22
@YSZN (Y1-Axis Line with Label & Value) 9-23
@YSZNL (Y1-Axis Line with Label & Value on Left) 9-24

Section 10: User-Defined Objects 10-1

@USER_CIRCLE (Outlined Circle) 10-2
@USER_CIRCLE_ABOVE (Outlined Circle above Chart Area) 10-4
@UF/@USER_FILL (Color-Filled Rectangle) 10-6
@USER_FILL2 (Pattern-Filled Rectangle) 10-7
@USER_FILL_CIRCLE (Color-Filled Circle) 10-8
@USER_FILL_CIRCLE_ABOVE (Color-Filled Circle above Chart Area) 10-9
@USER_FILL_CIRCLE2 (Pattern-Filled Circle) 10-10
@USER_FILL_CIRCLE2_ABOVE (Pattern-Filled Circle above Chart Area) . 10-11
@USER_MARKER (User-Defined Marker) 10-13
@USER_MARKER2 (User-Defined Marker with Value) 10-14
@USER_RECT (Outlined Rectangle) 10-15
@UW (User-Defined Vertical Band) 10-16
@WC (Color @UW Vertical Band) 10-17

Section 11: Chart Type Macros 11-1

@3DSCAT (3D Scatter Chart) 11-2
@COMBO (Combo Chart) 11-3
@COMPARE2 (2-Series Absolute Bar Chart) 11-4
@GANTT (Gantt Chart) 11-5
@GRAPHTYPE (Graph Type) 11-6
@PARETO (Pareto Chart) 11-10
@POLAR (Polar Chart) 11-11
@RIVER (Floating Area Series) 11-12

Section 12: Box Plot Macros 12-1

@BP/BP1 (Box Plot with Square Tail) 12-2
@BP2 (Box Plot with T-Style Tail) 12-3
@BP3 (Box Plot with I-Style Tail) 12-4
@BPH (Horizontal Box Plot) 12-5
@BPW (Box Plot Width) 12-6
@IN (Move First Box Plot In) 12-7
@MC (Marker Colors for Box Plots) 12-8
@MK (Number of Markers) 12-9

@MS (Marker Shapes for Box Plots) 12-10
Section 13: Macros for Gauges 13-1
@GAUGE_ANGLE (Gauge Angle) 13-2
@GAUGE_BORDER_STYLE (Gauge Border Style) 13-3
@GAUGE_BORDER_THICKNESS (Gauge Border Thickness) 13-4
@GAUGE_CHART (Gauge Chart) 13-5
@GAUGE_MULTIPLE_NEEDLES (Multiple Needles) 13-6
@GAUGE_NEEDLE_STYLE (Gauge Needle Style) 13-7
@GAUGE_RANGE_COLOR (Color Gauge Bands) 13-10
@GAUGE_RANGE_START_STOP (Gauge Band Start/Stop) 13-11
@GAUGE_RANGE_THICKNESS (Gauge Band Thickness) 13-12
@GAUGE_RANGE_THRESHOLD (Gauge Band Threshold) 13-13
Section 14: Pie Chart Macros 14-1
@DATATEXT_PIE (Data Text Mode for Pie Charts) 14-2
@PIE_ROTATE (Pie Rotation Start Point) 14-3
@SMART_PIE_LABELS (Smart Pie Labels) 14-4
Section 15: Macros for Waterfall Charts 15-1
@WATERFALL (Waterfall Chart) 15-2
@WATERFALL2 (Waterfall Chart with Total Group) 15-3
@WF_CENTERTEXT (Center Waterfall Data Text) 15-4
@WF_CONNECT (Waterfall Feeler Lines) 15-5
Section 16: Miscellaneous 16-1
@EB (Error Bars) 16-2
@FONTANGLE (Font Angle) 16-3
@FONTNAME (Font Name) 16-4
@FONTSIZE (Font Size) 16-5
@FRAME (Frame Size/Location) 16-6
@HAT (Hat on Error Bars) 16-7
@LEGEND (Legend Size/Location) 16-8
@LEGEND_ORDER (Force Legend Order) 16-9
@LEGEND_WRAP_WIDTH (Legend Wrap Width) 16-10
@ORD_SPACE (Line Chart Drawing Mode) 16-11
@USER_LABEL_FONT (User-Defined Label Font) 16-12
Section 17: Troubleshooting 17-1
@DEBUG (Show Debug Information) 17-1
@PARAM_COUNT_FIXUP (Debug Marco) 17-1
@PARAM_FIXUP (Parameter Fix-Up) 17-1
@RESET (Reset Internal Data Range) 17-2
Appendix A: Alpha Index A-1

Section 1: Overview

Charts Unlimited is a replacement charting library for IBM Cognos Series 7 Impromptu. It adds significant new charting capabilities for Series 7 designers and developers:

- Box Plots
- Gauge Charts
- Waterfall Charts
- Pareto Charts
- Polar Charts
- 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
- Expanded Correlation Charts, with any number of series formatted as bar/line/area (@COMBO)
- Runtime editing functions for most chart properties
- Drop Shadow and Alpha Channel transparency effects
- Ability to use any field or function as an input parameter to macros
- High-Quality, Anti-Aliased Rendering
- New Bevel Drawing Effects (@BEVEL)
- Bezier Curves on Line and Area Charts (@CURVED_LINES)
- Fine-Tuned Control of Automatic Chart Layout (@AA)
- 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.

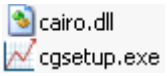
System Requirements

In order to use these macros, your system must be equipped with:

- Microsoft Windows XP, NT, 2000 or Vista
- Cognos Series 7 Impromptu

Installation & Setup

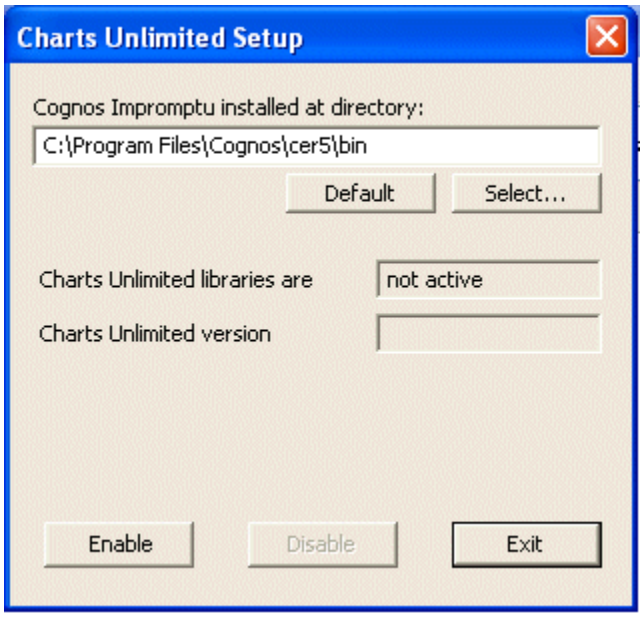
The ***Charts Unlimited*** library consists of two library files: cairo.dll and pgsdk32.dll. When you download a demo or the full version of ***Charts Unlimited***, you will receive the following files:



Copy the cairo.dll file to the following location:

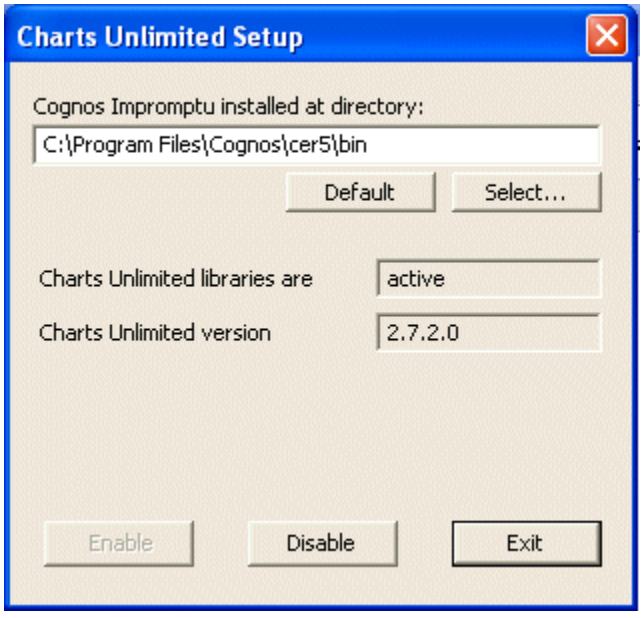
```
C:\Program Files\Cognos\cer5\bin\
```

Run the cgsetup.exe program to enable/disable the ***Charts Unlimited*** library (pgsdk32.dll).



- Click the Enable button to install or enable a previously disabled version of ***Charts Unlimited***.
- Click the Exit button to exit the setup application.

After ***Charts Unlimited*** is installed, the setup program can be used to restore the original Series 7 charting library.

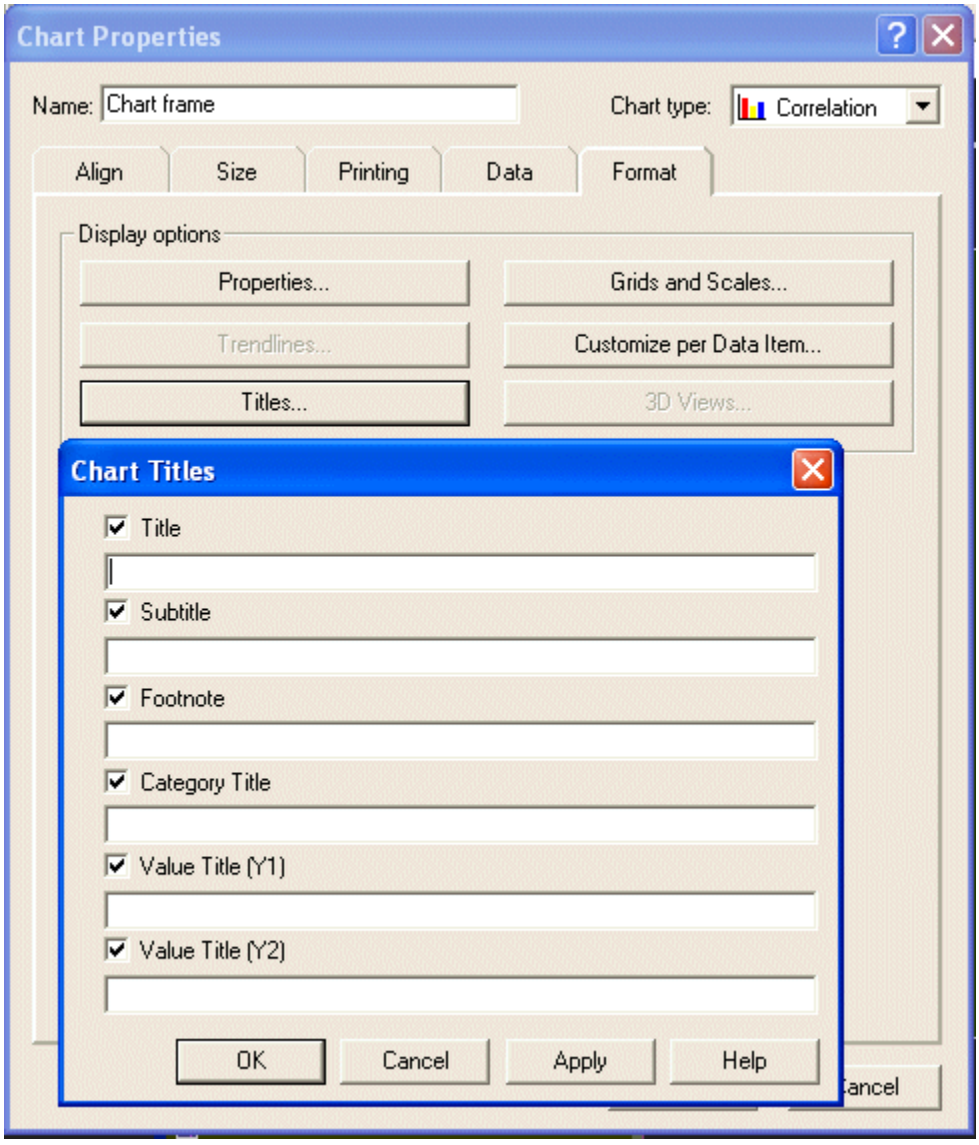


- Click the Disable button to restore the original pgsdk32.dll.
- Click the Exit button to exit the setup application.

Getting Started

Charts Unlimited macros can be inserted in any chart title field (i.e., title, subtitle, footnote, etc.) field in a chart. Chart titles can be defined in the Chart Titles dialog that is available from the Chart Properties dialog in Cognos Impromptu Administrator.

- Right-click on a chart background and choose Properties to show the Chart Properties dialog.
- Click the Format tab in the Chart Properties dialog.
- Click the Titles button in the Format tab to show the Chart Titles dialog.



Even though macros are defined in the chart title fields, they will not appear as text in your chart. If you need to use a particular title field to define a chart title and to enter a macro, 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
```

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_LABEL_FONT 14 255 0 0 255 Splash
```

Multiple macros may be defined in a single title field. When multiple macros are used, separate each macro with a space.

EXAMPLE:

```
@BP2 @MK 8
```

If the macro includes a string and multiple macros are defined in the same title field, a tilde (~) character must be appended 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
```

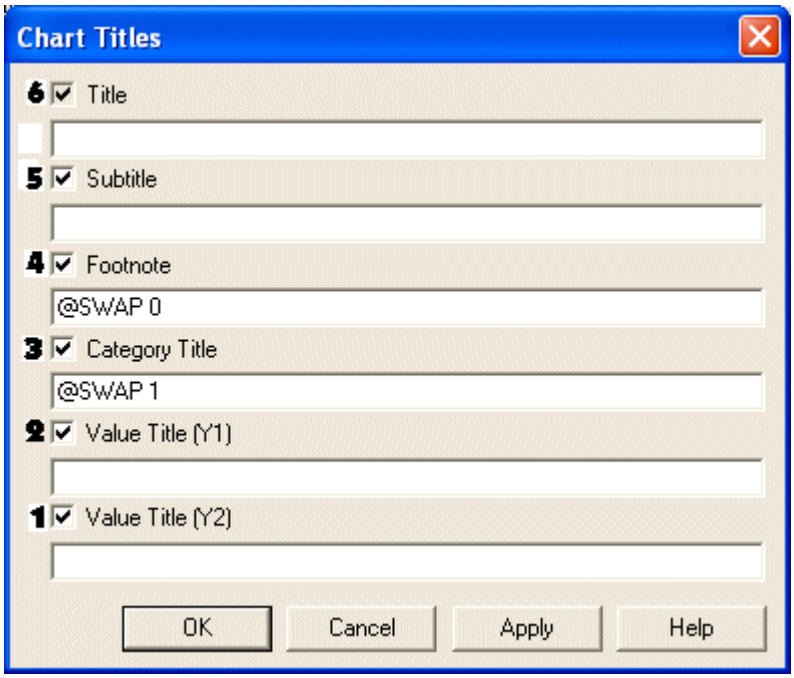
For macros that can be applied to a particular series in a chart, the *nSeries* parameter defines the series to which the macro is applied. In most cases, the *nSeries* parameter can be assigned a value in the range: minus one (-1)...*n* (where: *n* is the total number of series in the chart). For these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

Order of Evaluation

Because chart enhancement macros can be defined in any chart title field (i.e., Y2 Axis Title, Y1 Axis Title, X Axis Title, Footnote, Subtitle, or Title), you could potentially define conflicting macros. To avoid this potential conflict, the macros are evaluated in the following order:

1. Value Title (Y2)
2. Value Title (Y1)
3. Category (or X-Axis) Title
4. Footnote
5. Subtitle
6. Title

For example, assume "@SWAP 0" is defined in the Footnote field and "@SWAP 1" is defined in the Group Title field.



In this example, the "@SWAP 0" macro would be used because it is evaluated after the "@SWAP 1" definition.

If multiple macros are defined in the same title field, macros are parsed from left-to-right. For example, assume the following macros are defined in the Footnote field:

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

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

Persistence

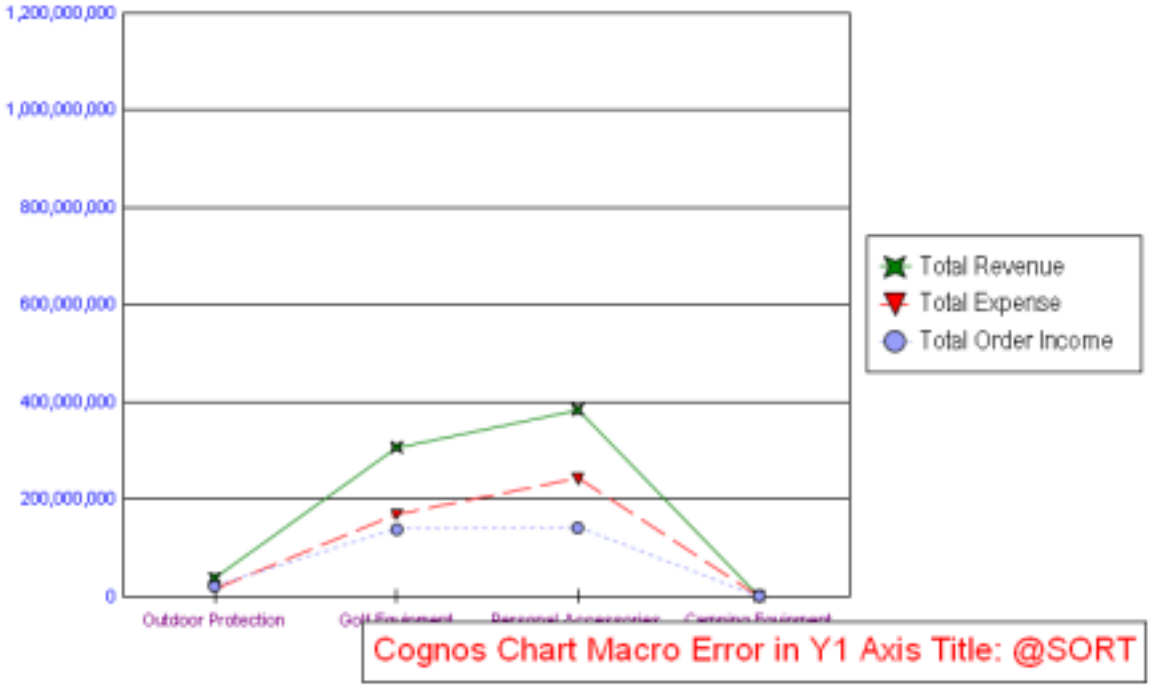
Many of the chart enhancement macros set a property of the chart and that property will remain set, even if the macro is removed from the field. For example, if "@SWAP 1" is used to reverse the series/groups in the chart, the series/groups will remain reversed even if the "@SWAP 1" macro is removed. 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, @XY, etc.) 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 in this document identifies whether or not the macro is persistent.

Error Reporting

If your macro encounters an error, a message will be displayed in the footnote area of the chart. Example:



The error message indicates the field (e.g., Group Title, Footnote, Subtitle, Title, etc.) and the macro where the error occurred. The most common errors are:

1) missing parameter(s)

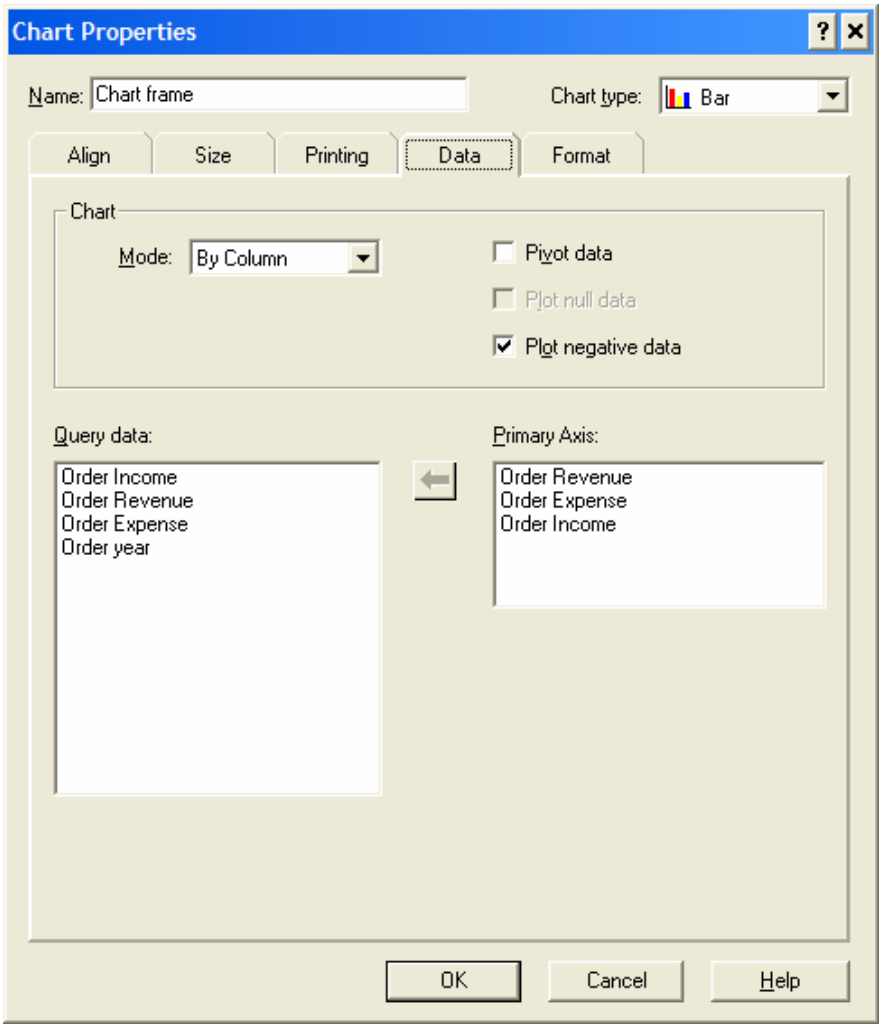
2) parameter is assigned an out-of-range value

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

Using Fields/Functions in Macros

You can use any field or function in an Impromptu report as an input parameter to a ***Charts Unlimited*** macro.

- Add the report field or function that you want to use with a macro to the end of “Primary Axis” list in the Data tab of the Chart Properties dialog.



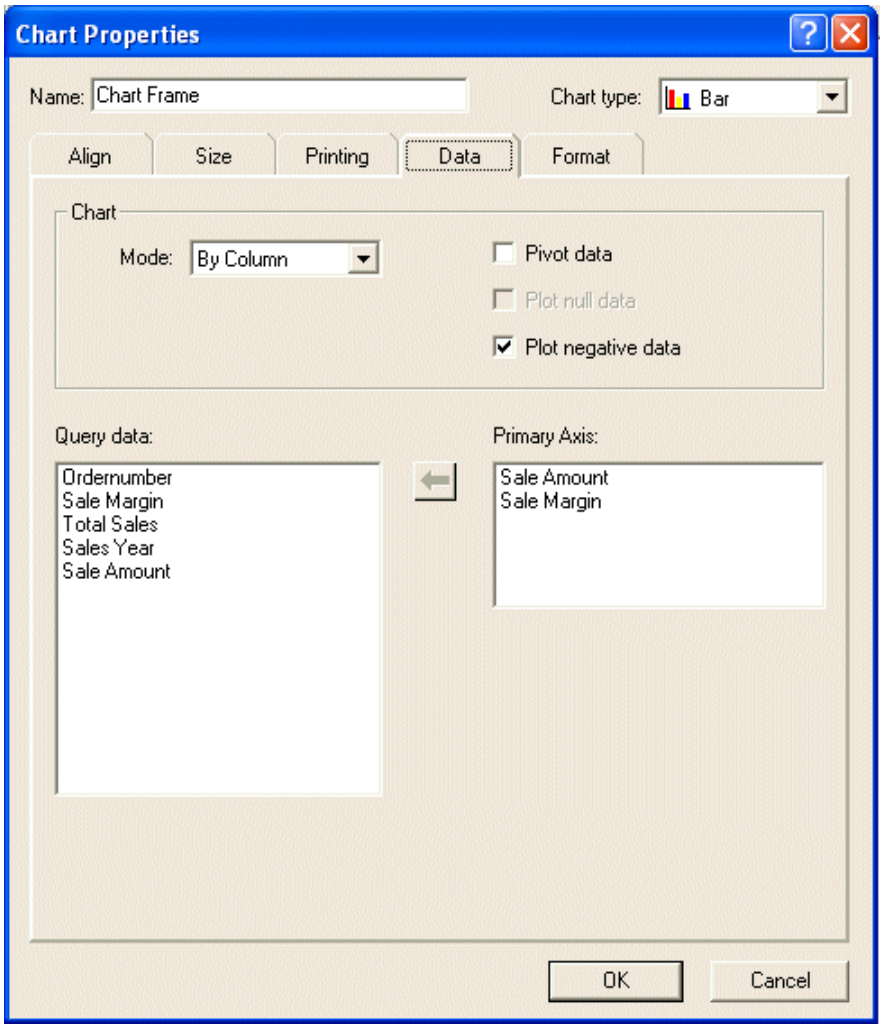
- Make a note of the zero-based position of this field in the “Primary Axis” list.
- In your macro, use the letter “P” followed by the number that represents the position in the “Primary Axis” list.

IMPORTANT NOTES:

- Do not reference an item in the “Primary Axis” list that is part of the dataset being graphed (i.e., do not use "P0"). This would cause that particular data series to disappear.
- Do not use the same “P” parameter twice (e.g., @CXY 0 1 P5 P5 is not valid).
- Only NUMERIC (integer, Boolean, real) parameters can be mapped. ***Charts Unlimited*** does not currently support runtime mapping of string parameters.

EXAMPLE:

In the Data tab of the Chart Properties dialog, the "Sale Margin" field is added at zero-based position 1 in the "Primary Axis" list.



The following macro will create a line on the Y1-axis at the value of "Sale Margin".

```
@Y P1
```

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
- @AA3; Automatically arrange objects in a chart with Frame & Legend adjustment.
- @AA4; Automatically arrange objects in a chart with Maximized Frame adjustment.

@AA (Auto Arrange)

This macro automatically arranges elements in a chart.

SYNTAX:

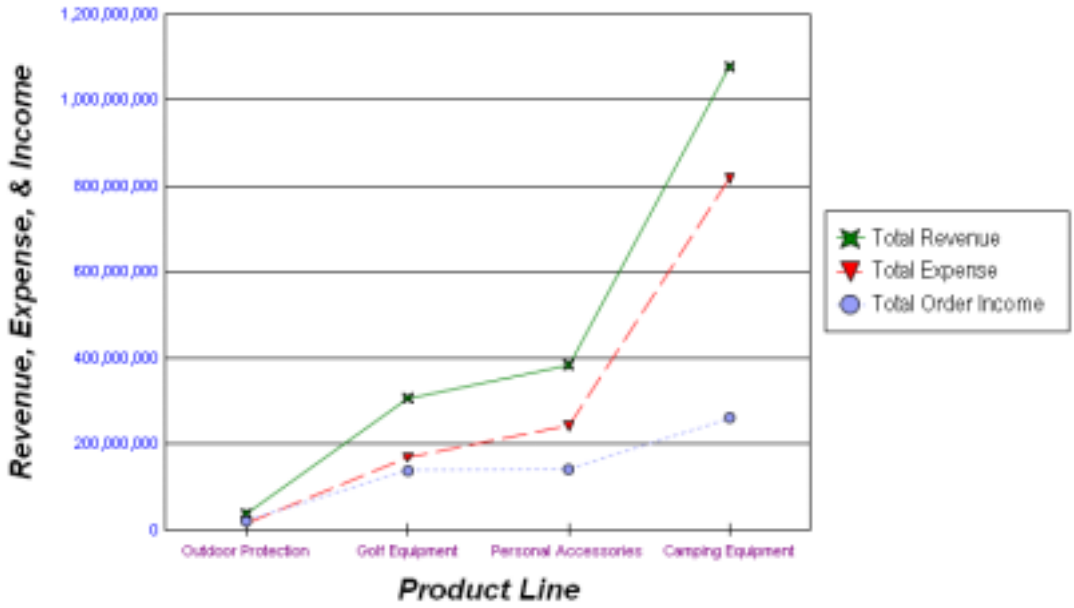
```
@AA
```

PARAMETERS:

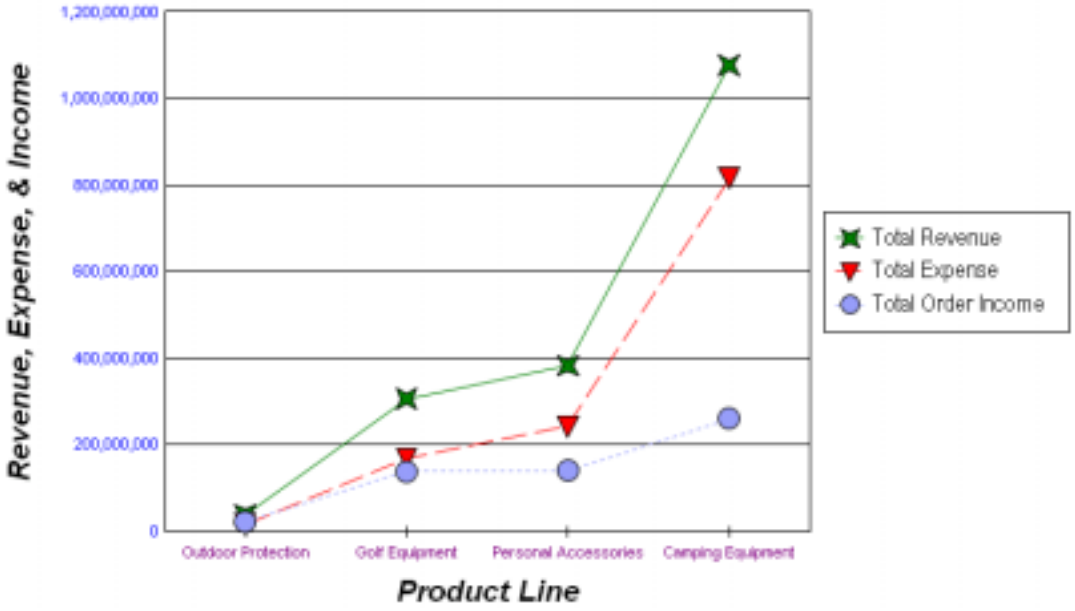
None

EXAMPLE:

```
Chart 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:

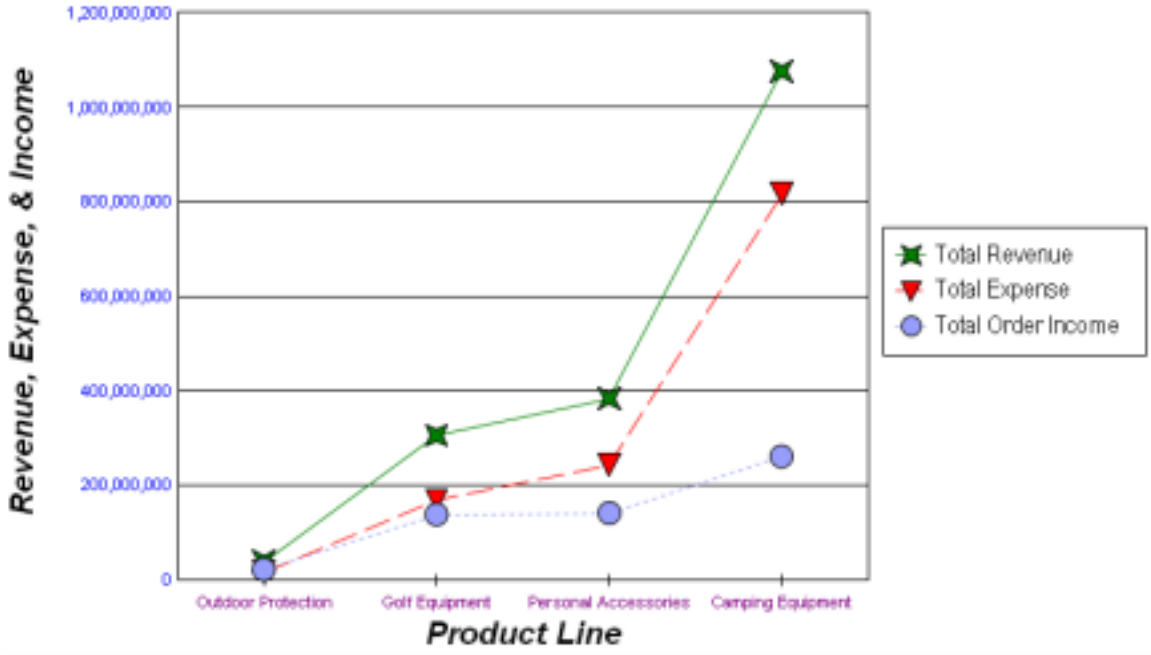
```
@AA2 nFixup
```

PARAMETERS:

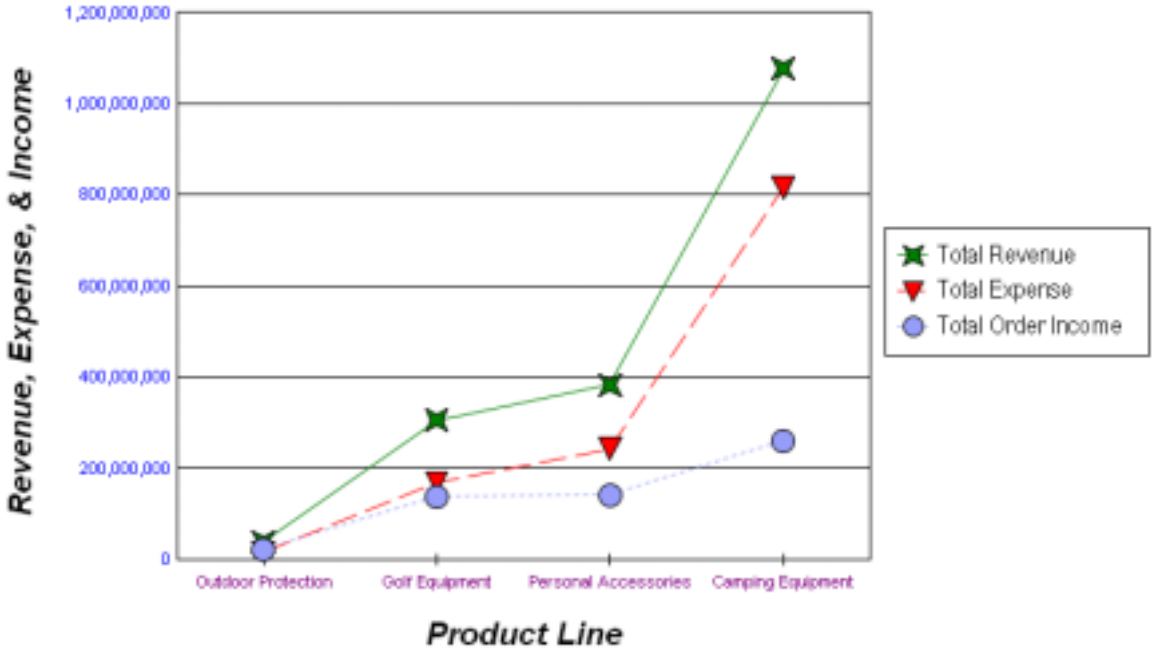
nFixup; -16000...16000 specifies how much to move the bottom of the chart frame up (negative values) or down (positive values).

EXAMPLE:

```
@AA2 500
```




```
@AA2 -500
```



PERSISTENT:

NO

@AA3 (Auto Arrange with Frame & Legend 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 it 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

3) The *nLegendMode* parameter can be used to force the legend to a specific location.

SYNTAX:

```
@AA3 nFixup nLegendMode
```

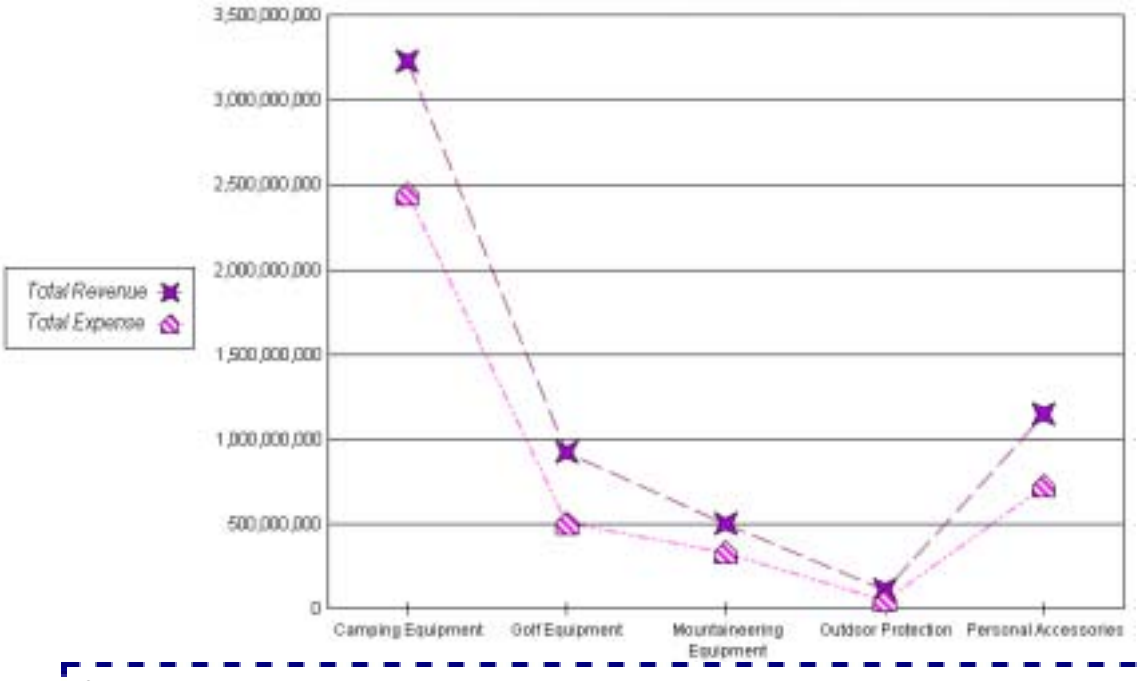
PARAMETERS:

nFixup; -16000...16000 specifies how much to move the bottom of the chart frame up (negative values) or down (positive values).

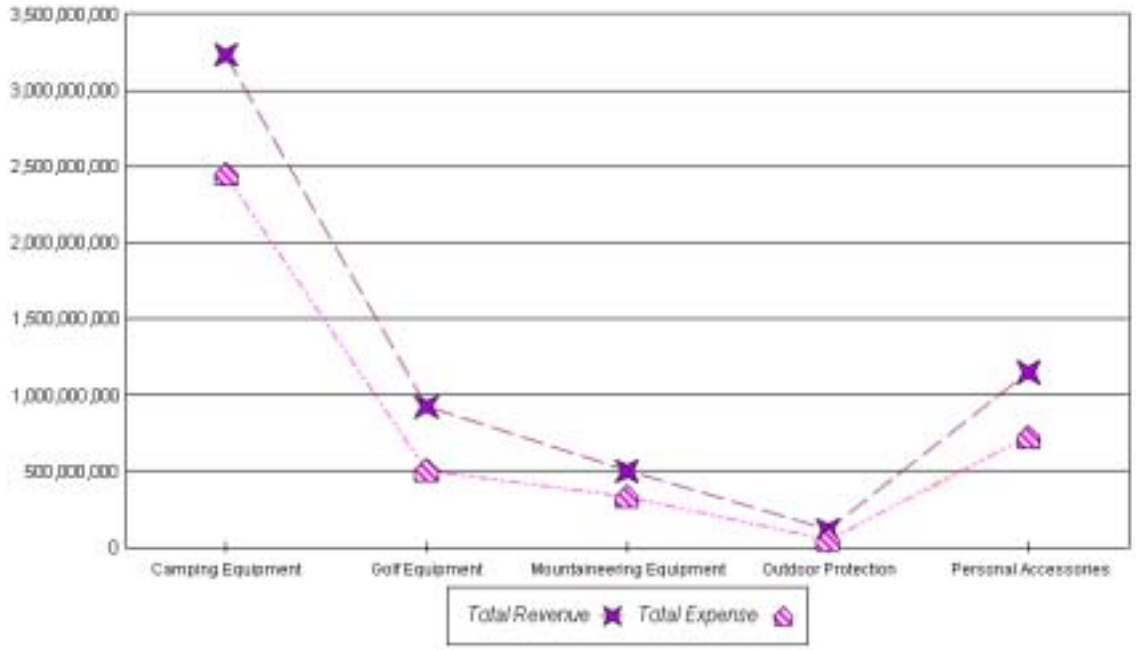
nLegendMode; 0...2 specifies the position to force the legend (0=right, 1=left, 2=bottom)

EXAMPLE:

```
@AA3 500 1
```




```
@AA3 -500 2
```



PERSISTENT:

NO

@AA4 (Auto Arrange with Maximized Frame Adjustment)

This macro is the same as @AA2 except it more accurately maximizes the chart frame when there are no titles. 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/Group labels

SYNTAX:

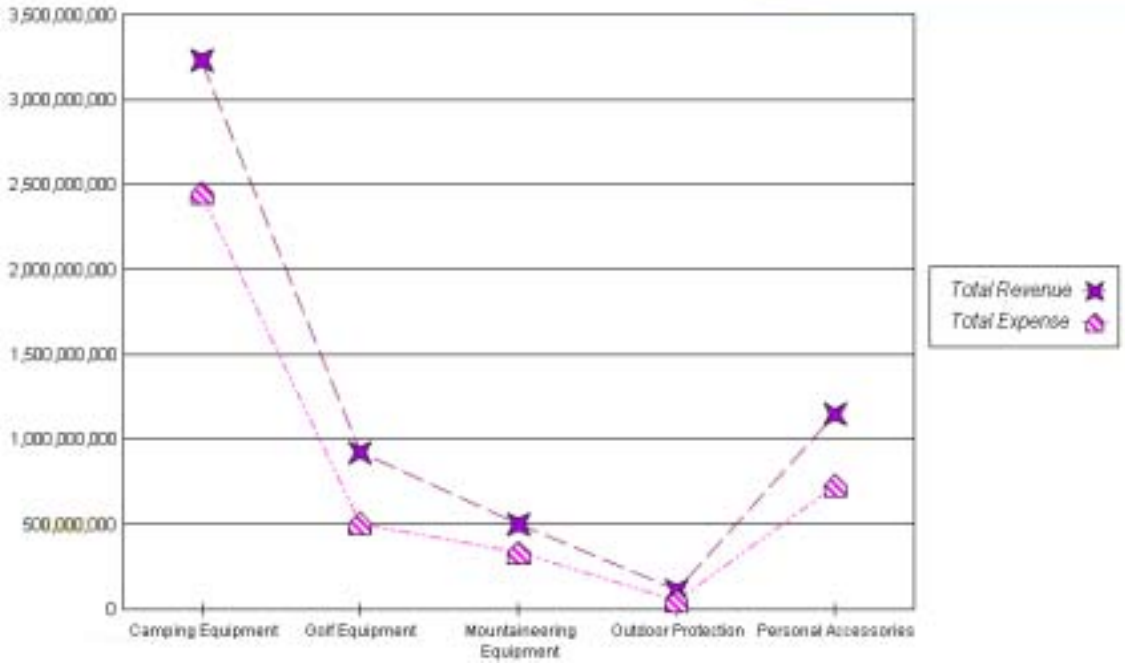
```
@AA4 nFixup
```

PARAMETERS:

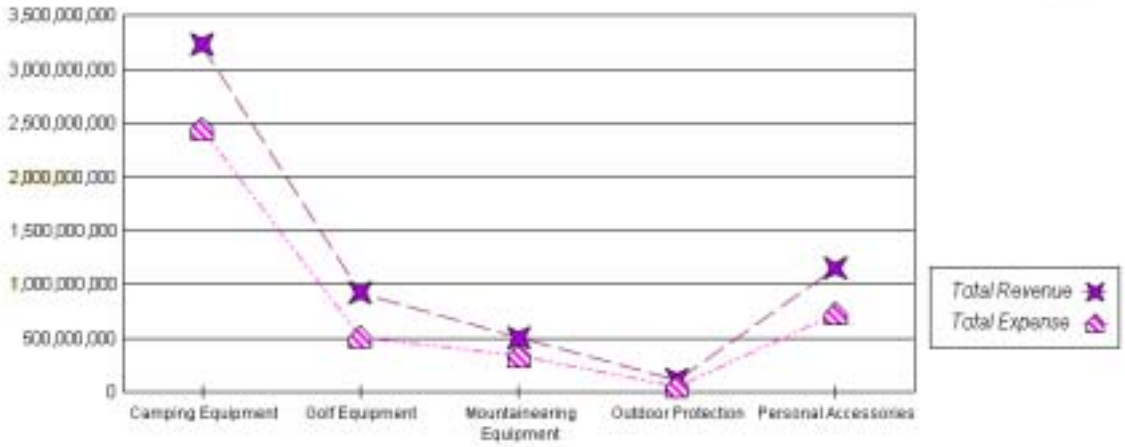
nFixup; -16000...16000 specifies how much to move the bottom of the chart frame up (negative values) or down (positive values).

EXAMPLE:

```
@AA4 500
```



```
@AA4 -10000
```



PERSISTENT:

NO

Section 3: Axis Macros

These macros can be used to control the appearance of the axes in a chart:

- @AXIS; Assign a Series to an Axis
- @DX; X-Axis Divisions
- @DY; Y-Axis Divisions
- @DY2; Y2-Axis Divisions
- @GX; X-Axis Grid Style
- @GY; Y-Axis Grid Style
- @HOUR_SCALE; Draw minor axis hour scale on a time axis
- @PX; X-Axis Precision
- @PY; Y1-Axis Precision
- @PY2; Y2-Axis Precision
- @SC; Y-Axis Scale
- @SCX; X-Axis Scale
- @SCY2; Y2-Axis Scale
- @X_AXIS_MODE; X-Axis Mode
- @Y1_FORCE_PERCENT; Y1-Axis force Percent Format
- @Y1_FORCE_PERCENT2; Y1-Axis force Percent/Numeric Format
- @Y1_INVERT; Y1-Axis Invert
- @Y1BASE; Y1-Axis Base Line
- @Y2_FORCE_PERCENT; Y2-Axis force Percent Format
- @Y2_INVERT; Y2-Axis Invert
- @Y2BASE; Y2-Axis Base Line
- @Y2SLAVE; Slave Y2-Axis to Y1-Axis
- @Y2SLAVE2; Slave Y1/Y2-axes to Max Value

@AXIS (Assign Series to Axis)

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

SYNTAX:

```
@AXIS nSeries bAxis
```

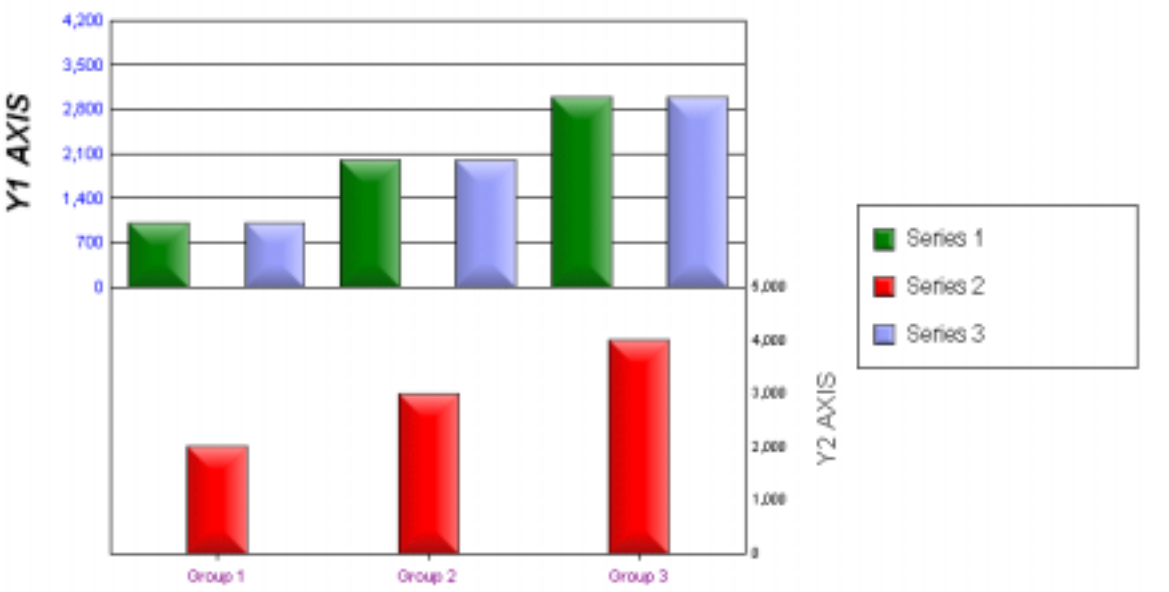
PARAMETERS:

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.

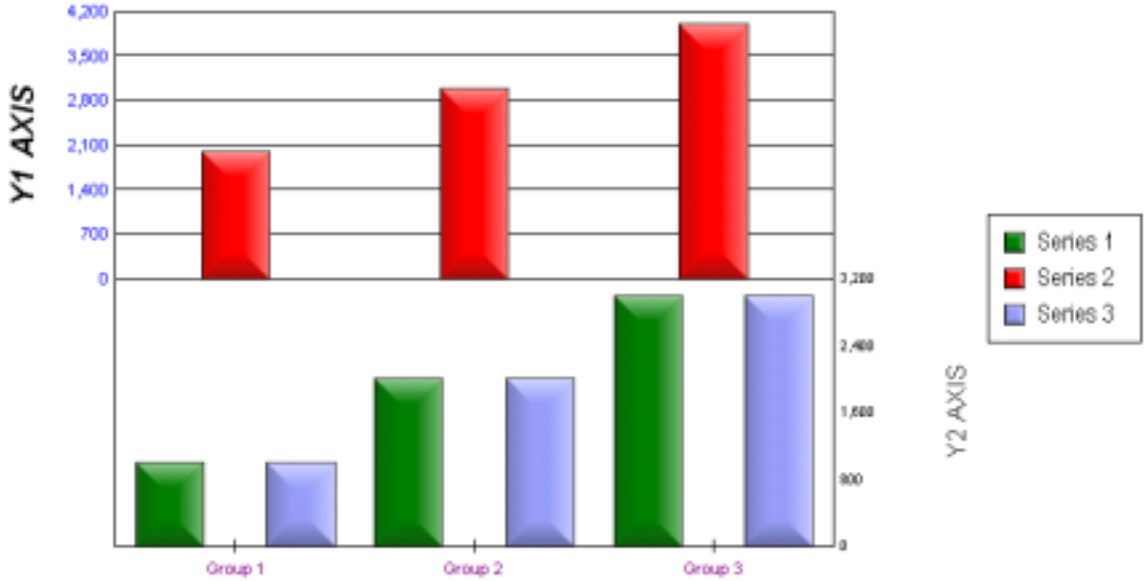
bAxis; 0=assign *nSeries* to Y1-Axis, 1=assign *nSeries* to Y2-Axis

EXAMPLE:

```
@AXIS 0 0 @AXIS 1 1 @AXIS 2 0
```




```
@AXIS 0 1 @AXIS 1 0 @AXIS 2 1
```



PERSISTENT:

YES

@DX (X-Axis Divisions)

This macro can be used to set 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:

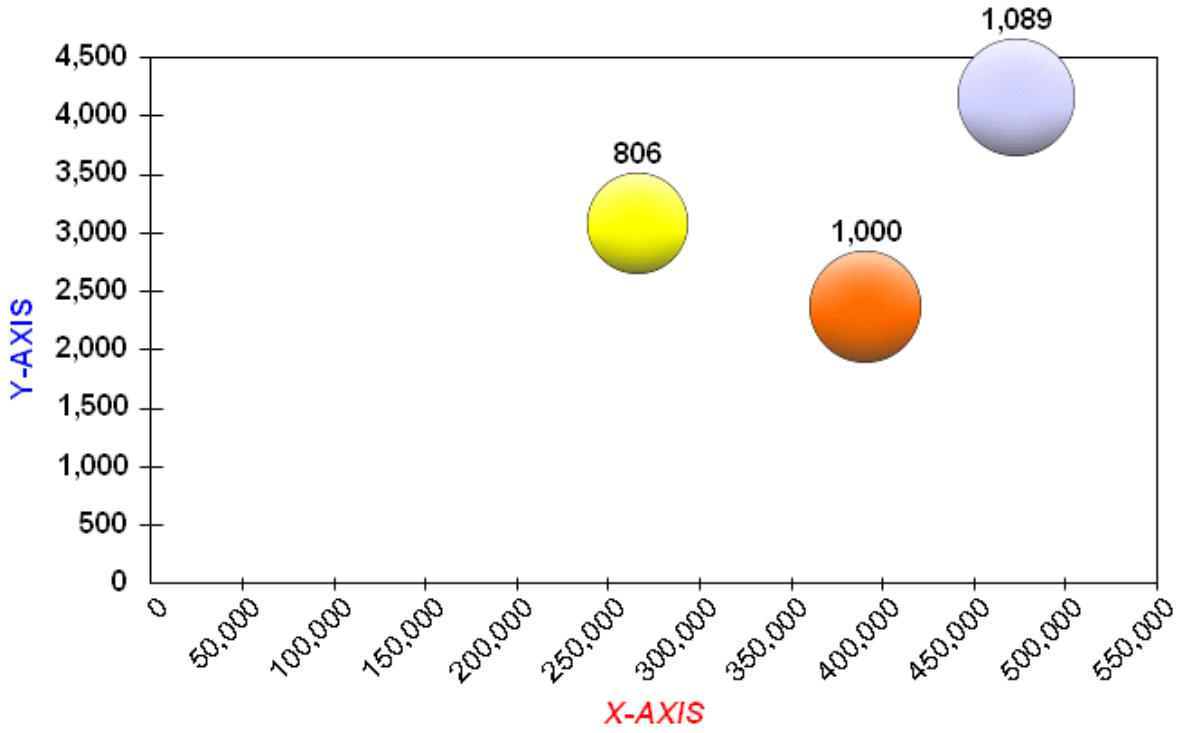
```
@DX nDivisions
```

PARAMETERS:

nDivisions; Number of divisions on the X-Axis

EXAMPLE:

```
@FONTANGLE 8 5
@DX 10
```



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 can be used to set the number of division on the Y-Axis.

SYNTAX:

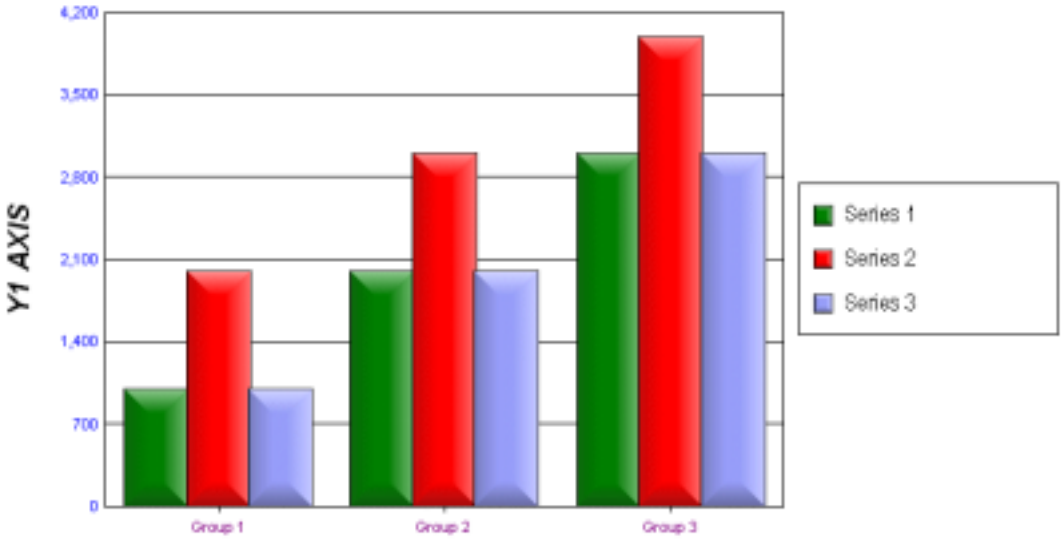
```
@DY nDivisions
```

PARAMETERS:

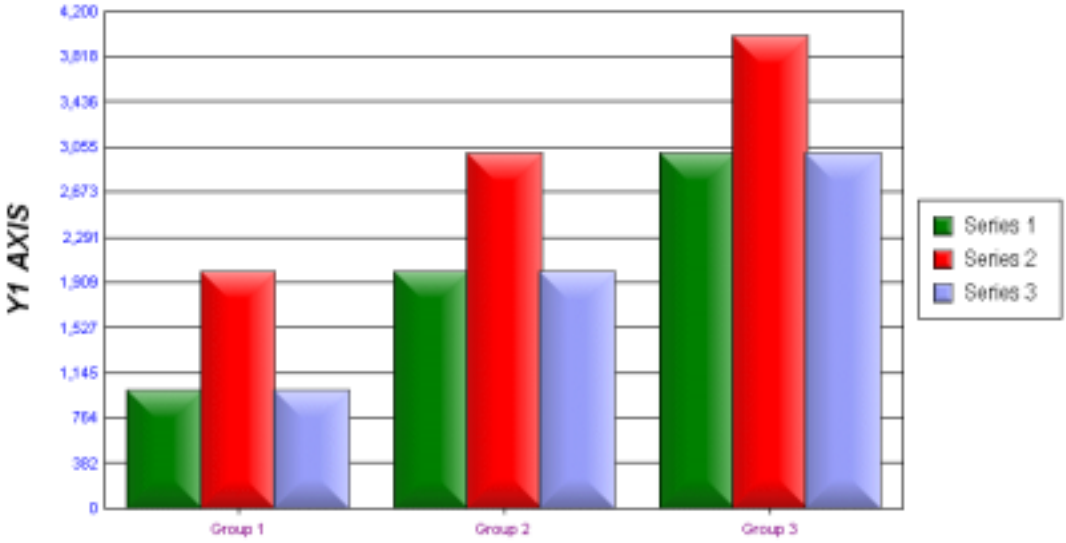
nDivisions; Number of divisions on the Y-Axis

EXAMPLE:

```
@DY 5
```




```
@DY 10
```



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 can be used to set the number of division on the Y2-Axis.

SYNTAX:

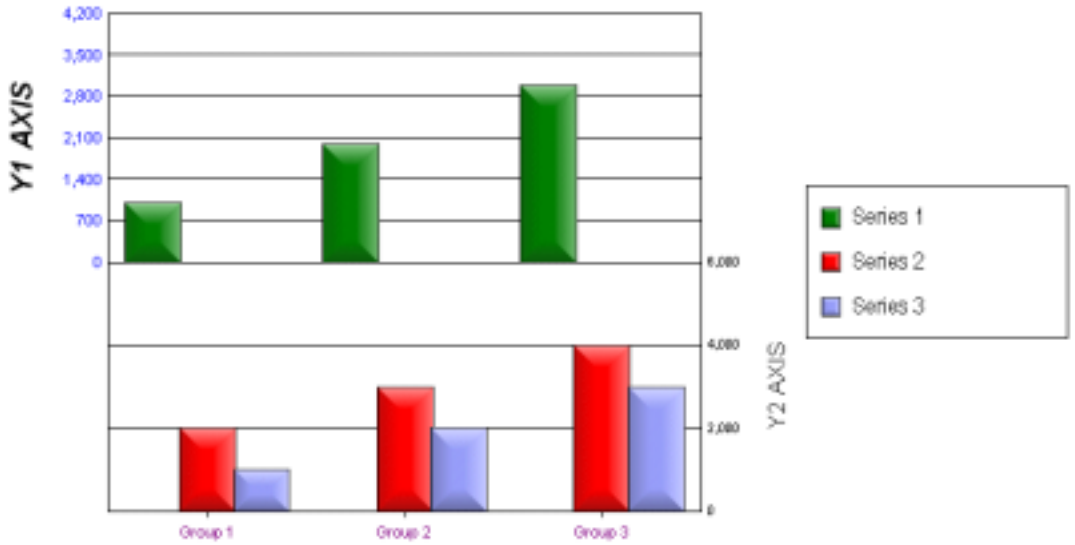
```
@DY2 nDivisions
```

PARAMETERS:

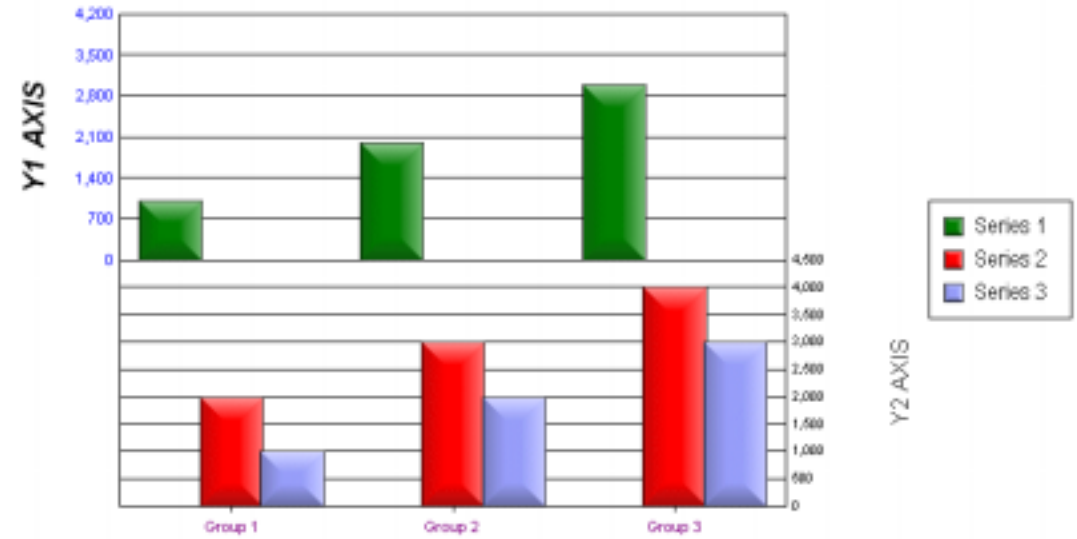
nDivisions; Number of divisions on the Y2-Axis

EXAMPLE:

```
@DY2 2
```




```
@DY2 8
```



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

@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 nGridStyle
```

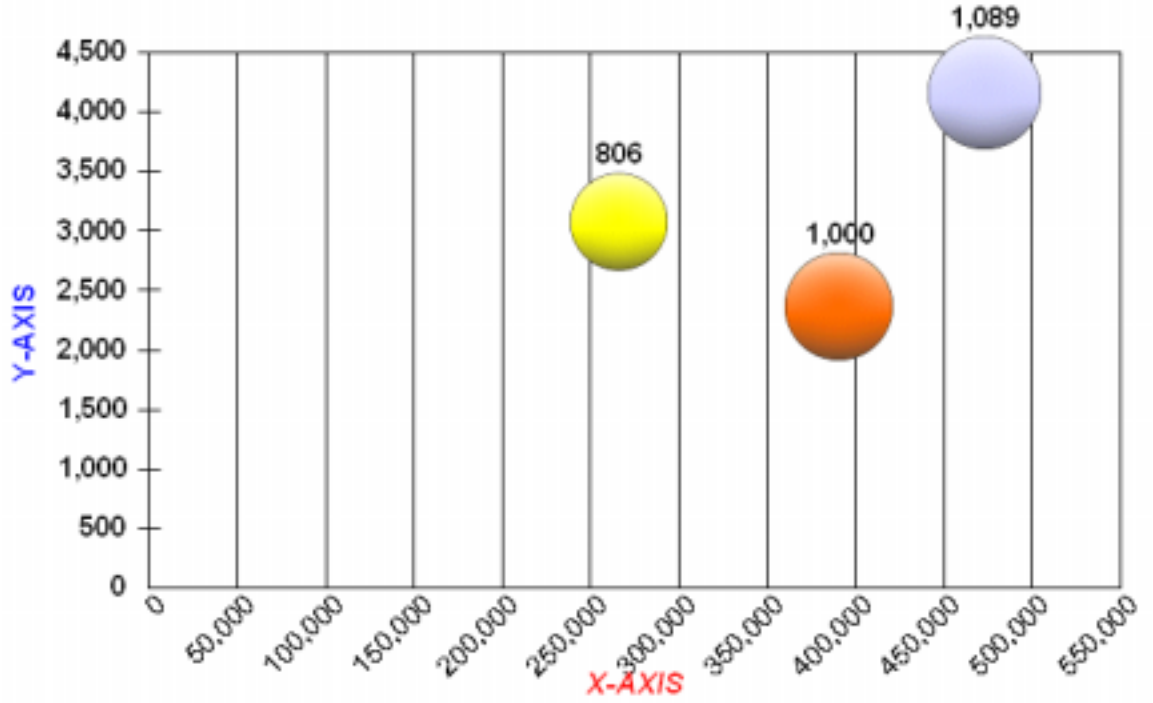
PARAMETERS:

nGridStyle; 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:

In this example line chart, the Group/O1-Axis Grid Style is set to a standard grid with no ticks.

```
@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 nGridStyle
```

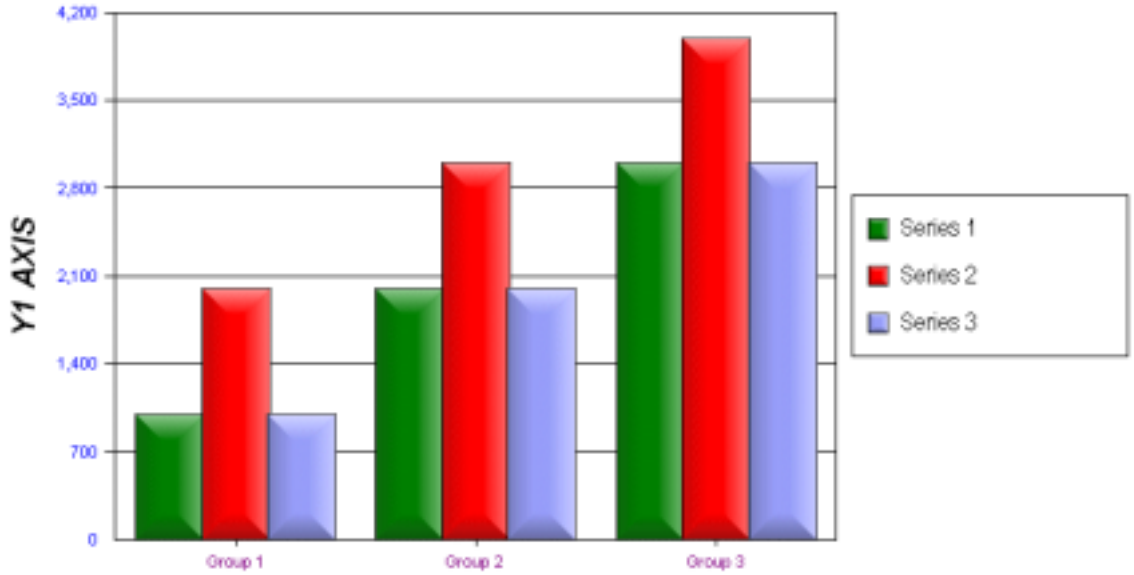
PARAMETERS:

nGridStyle; 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 2
```



PERSISTENT:

NO

ALSO SEE:

@GX

@HOUR_SCALE (Minor Axis Hour Scale)

When Y-Axis minor gridlines are enabled, this macro draws a minor axis hour scale on a time axis in a Gantt chart.

SYNTAX:

```
@HOUR_SCALE bActivateHours nInterval szHourFormat
```

PARAMETERS:

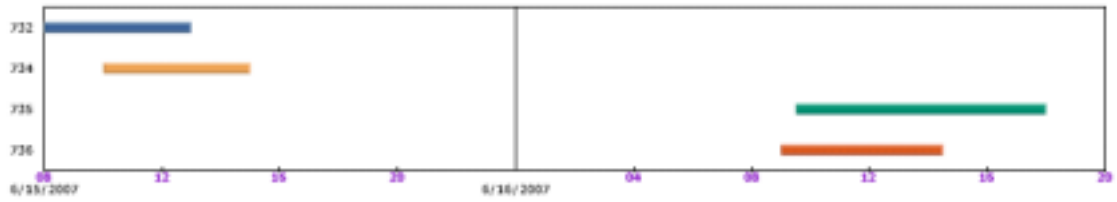
bActivateHours; 0=do not show hours, 1=show hours

nInterval; 1...24 interval to show hours. For example, a value of 6 shows a label at every sixth hour: 6AM, Noon, 6PM, etc.

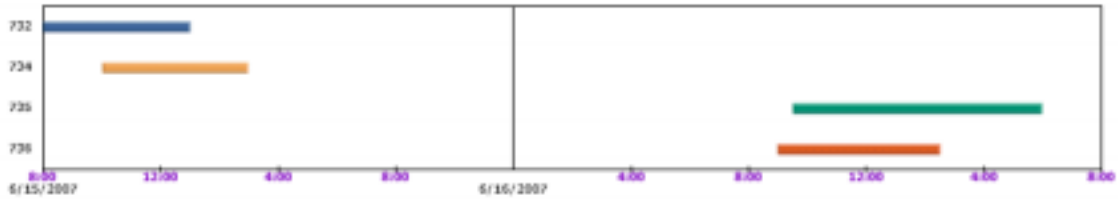
szHourFormat; Label format string. Use standard Windows syntax: h=12-hour AM/PM time, H=24-hour time, h:mm=hours:minutes (e.g., 3:00), etc.

EXAMPLE:

```
@HOUR_SCALE 1 4 HH~
```




```
@HOUR_SCALE 1 4 h:mm~
```



PERSISTENT:

NO

NOTES:

Y-Axis minor gridlines must be enabled.

@PX (X-Axis Precision)

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

SYNTAX:

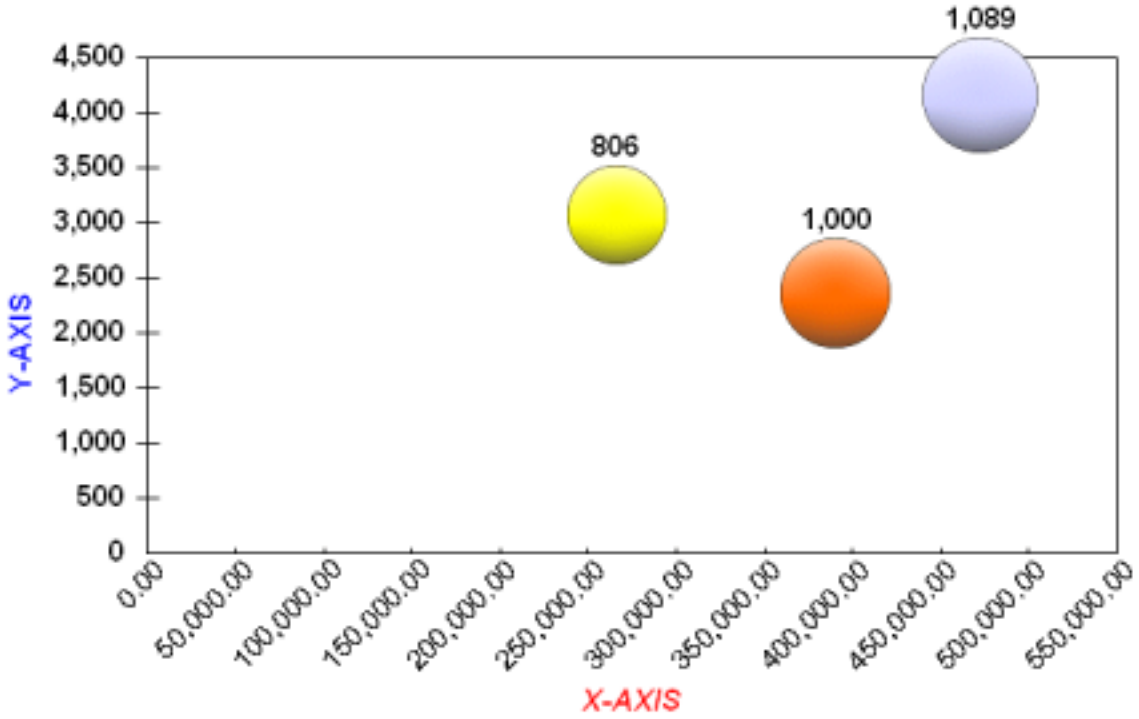
```
@PX nPrecision
```

PARAMETERS:

nPrecision; 0...9 specifies the number of decimal places for values on the X-Axis

EXAMPLE:

```
@PX 2
```



PERSISTENT:

YES

ALSO SEE:

@PY

@PY (Y1-Axis Precision)

This macro sets the decimal precision of values on the Y-axis.

SYNTAX:

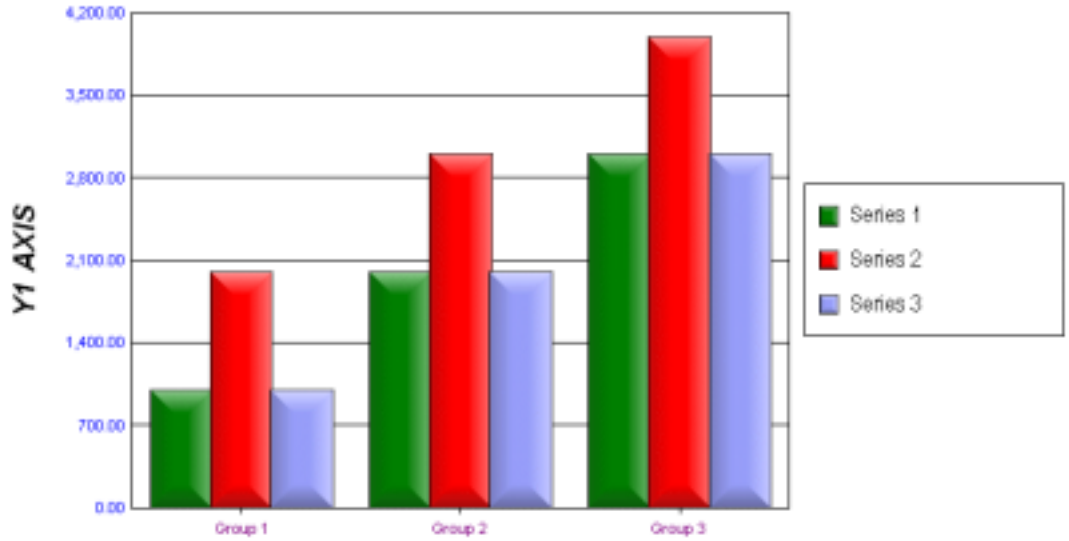
```
@PY nPrecision
```

PARAMETERS:

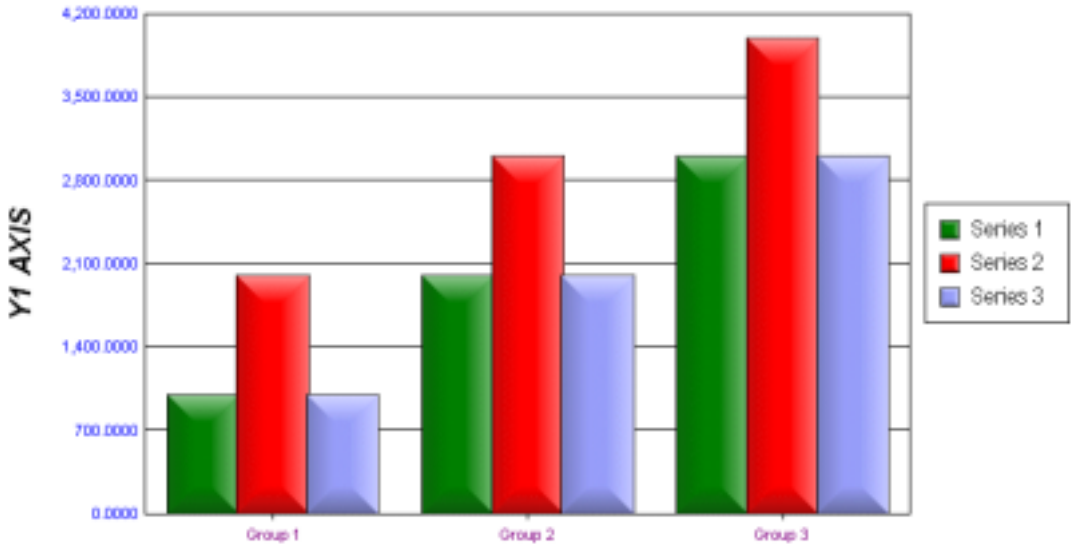
nPrecision; 0...9 specifies the number of decimal places for values on the Y-Axis

EXAMPLE:

```
@PY 2
```




```
@PY 4
```



PERSISTENT:

YES

ALSO SEE:

@PX, @PY2

@PY2 (Y2-Axis Precision)

In a dual-axis chart, this macro sets the decimal precision of values on the Y2-axis.

SYNTAX:

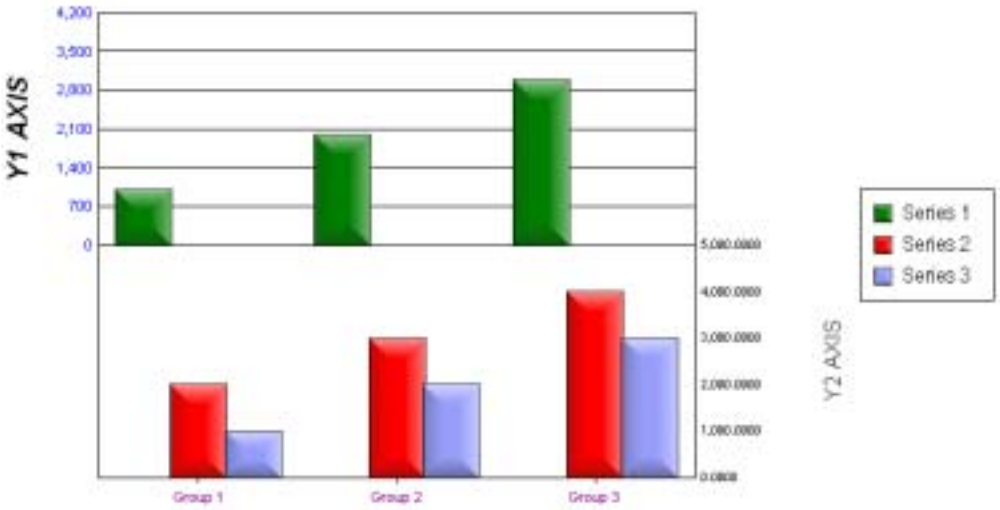
```
@PY2 nPrecision
```

PARAMETERS:

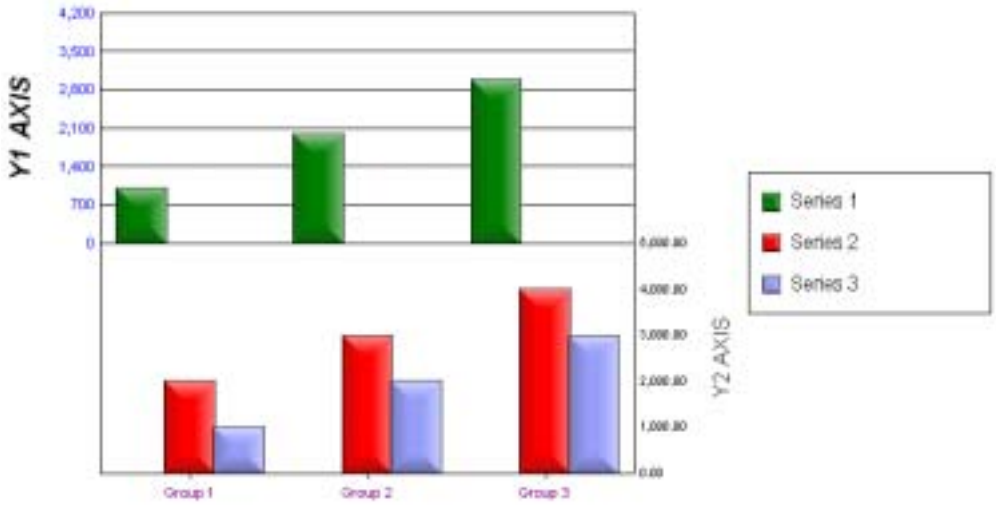
nPrecision; 0...9 specifies the number of decimal places for values on the Y2-Axis

EXAMPLE:

```
@PY2 4
```




```
@PY2 2
```



In these example charts, Y-axis values are shown on the left side of the chart and Y2-axis values are shown on the right side of the chart.

PERSISTENT:

YES

ALSO SEE:

@PX, @PY

@SC (Y-Axis Scale)

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

SYNTAX:

```
@SC fMin fMax
```

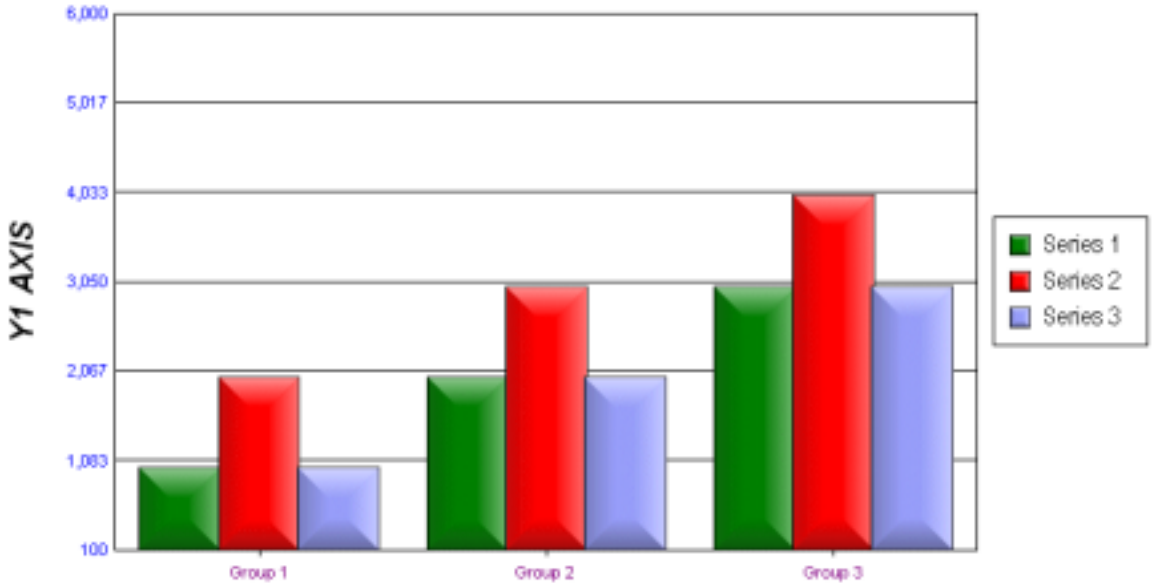
PARAMETERS:

fMin; Minimum value to show on the Y-Axis

fMax; Maximum value to show on the Y-Axis

EXAMPLE:

```
@SC 100 6000
```



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

@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
```

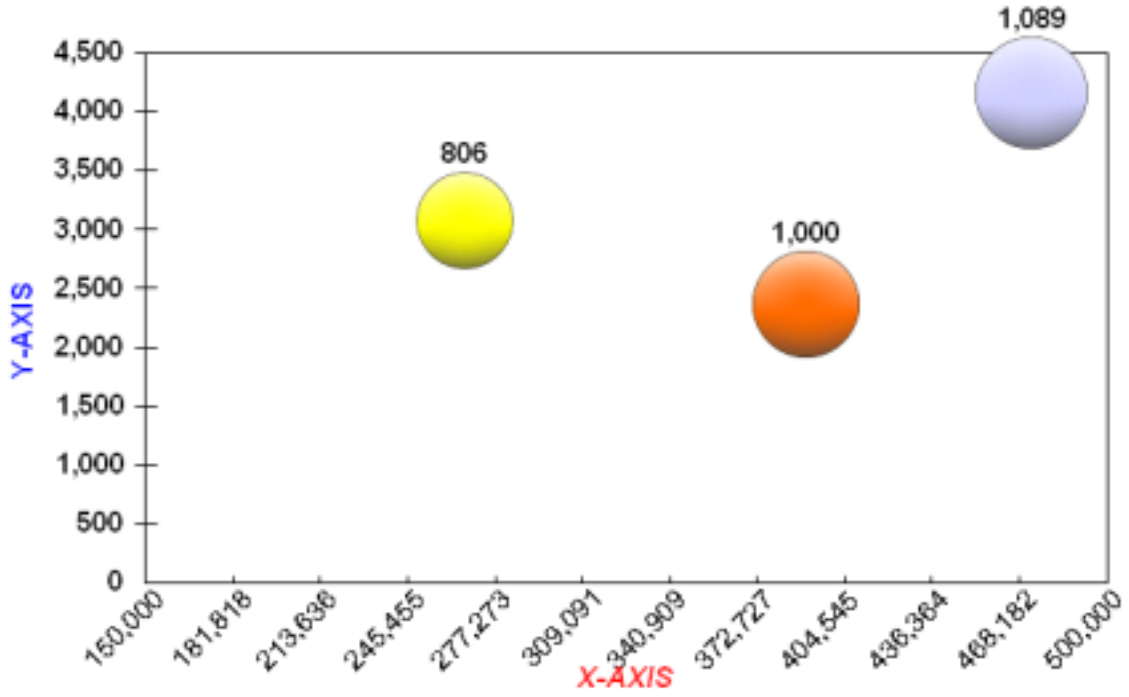
PARAMETERS:

fMin; Minimum value to show on the X-Axis

fMax; Maximum value to show on the X-Axis

EXAMPLE:

```
@SCX 150000 500000
```



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
```

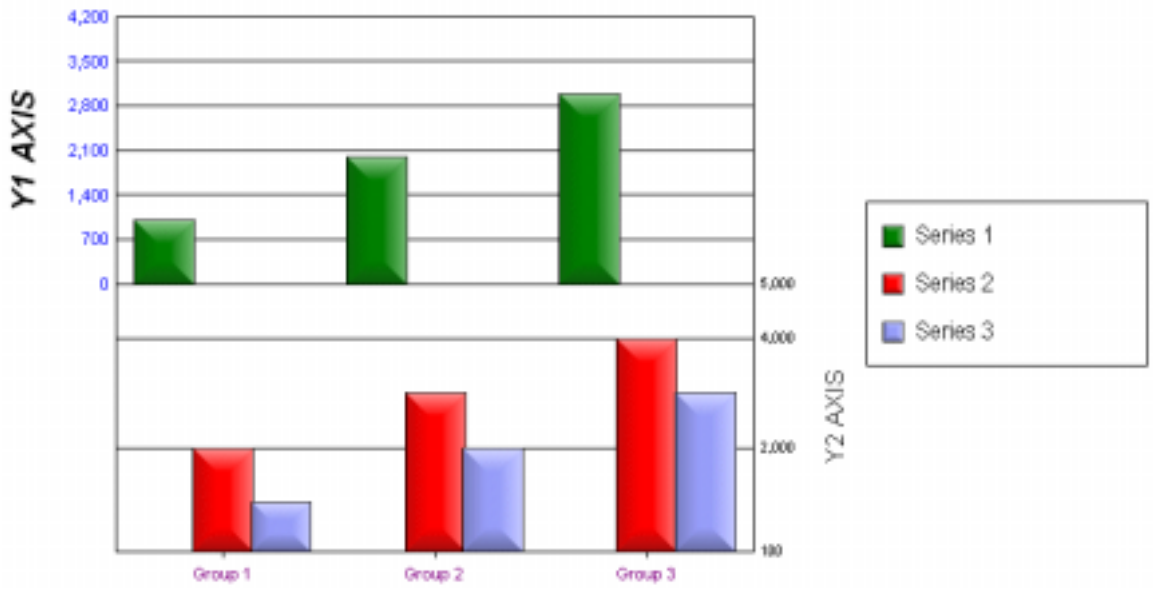
PARAMETERS:

fMin; Minimum value to show on the Y2-Axis

fMax; Maximum value to show on the Y2-Axis

EXAMPLE:

```
@SCY2 100 5000
```



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

@X_AXIS_MODE (X-Axis Mode)

This macro can be used to define an X-axis on a bar, line, or area chart that covers a specific range. It is specifically useful in a chart where the data is incomplete (i.e., it includes some but not all entries for a specified range). For example, the data may include entries for days 2, 3, 5, 14, 19, 20, and 30 in a given month. When this macro is not used, the chart will show seven group labels (2, 3, 5, 14, 19, 20, 30). If this macro is used to define groups 1 through 30, the chart will show 30 group/day labels with the seven defined groups/values in the correct position for each day.

SYNTAX:

```
@X_AXIS_MODE nMode nStart nStop
```

PARAMETERS:

nMode; 0

nStart; Starting group label string

nStop; Ending group label string.

PERSISTENT:

NO

NOTES:

- Label strings must be digits (e.g., 1999, 2000, 2001, etc.) that can be converted to integer values.
- This macro will not work if group labels are aliased with the @AGL macro.

@Y1_FORCE_PERCENT (Y1-Axis Percent Format)

This macro can be used to reformat Y1-axis labels using percent format.

SYNTAX:

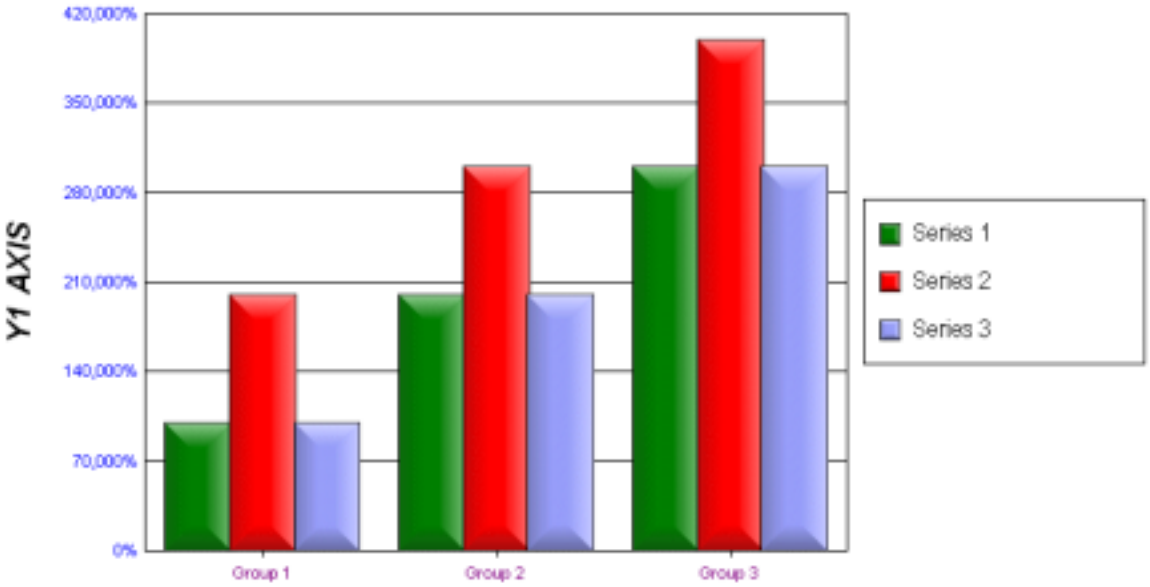
```
@Y1_FORCE_PERCENT bForce
```

PARAMETERS:

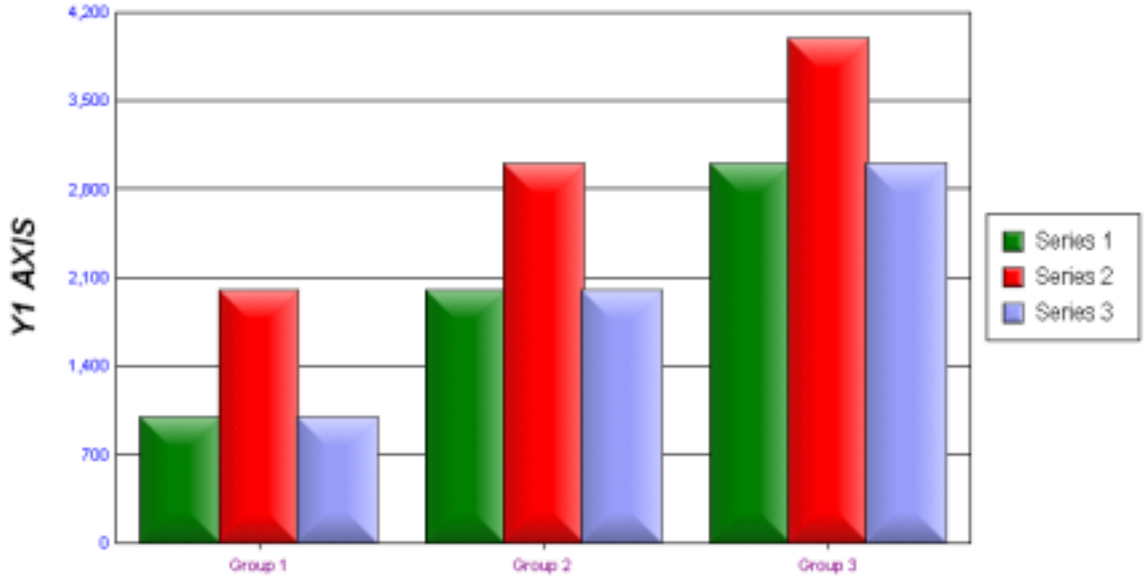
bForce; 0=Draw Y1-axis labels normally. 1=Use percent format.

EXAMPLE:

```
@Y1_FORCE_PERCENT 1
```




```
@Y1_FORCE_PERCENT 0
```



PERSISTENT:

NO

@Y1_FORCE_PERCENT2 (Y1-Axis Percent/Numeric Format)

This macro can be used to reformat Y1-axis labels to use percentage or numeric format.

SYNTAX:

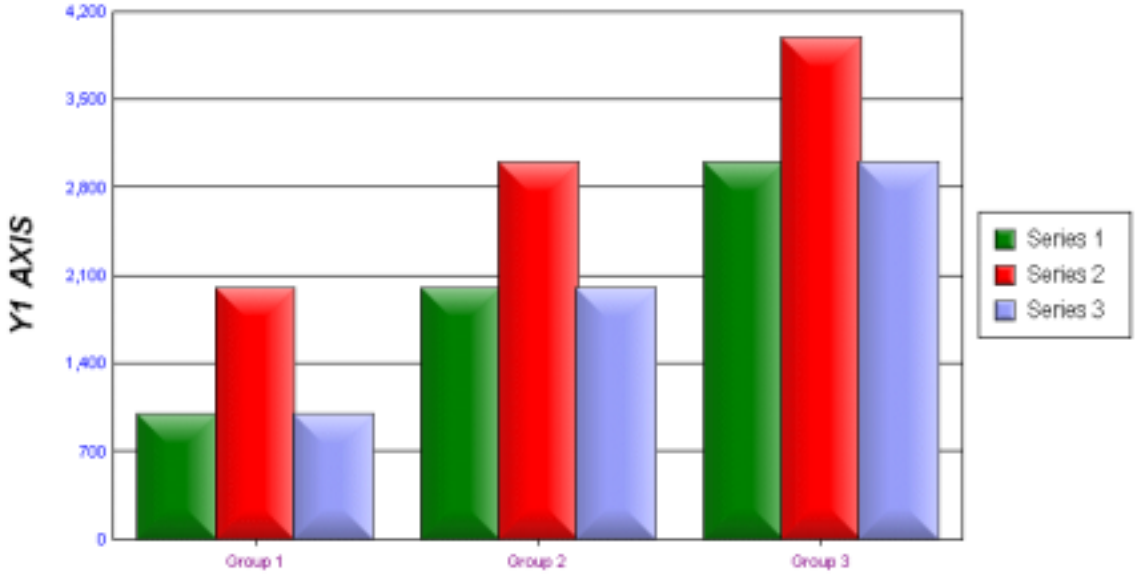
```
@Y1_FORCE_PERCENT2 bForce
```

PARAMETERS:

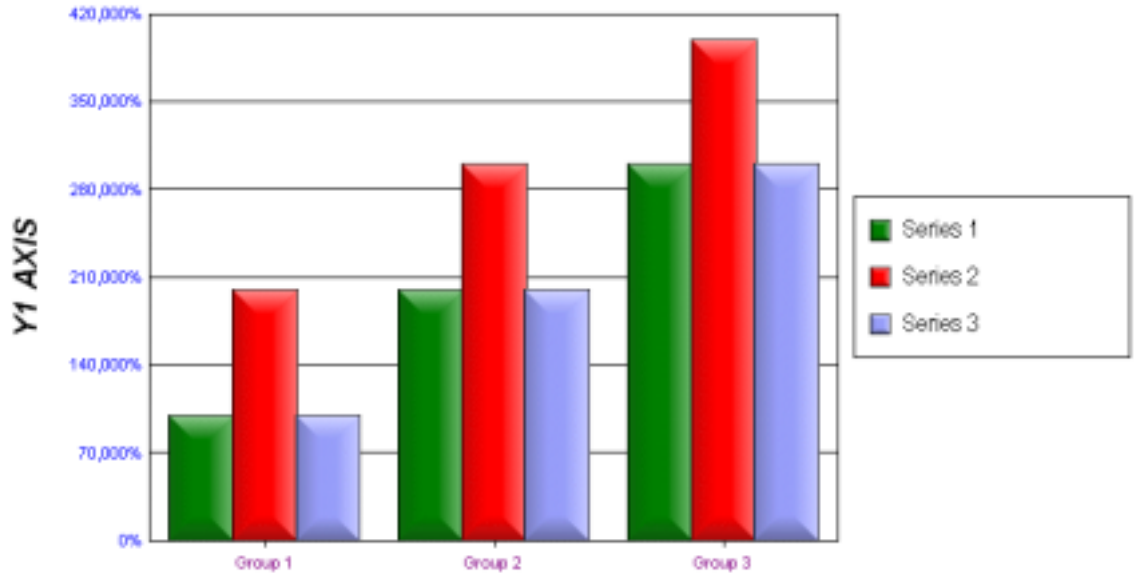
bForce; 0=Numeric Format. 1=Percent Format.

EXAMPLE:

```
@Y1_FORCE_PERCENT2 0
```




```
@Y1_FORCE_PERCENT2 1
```



PERSISTENT:

NO

@Y1_INVERT (Y1-Axis Invert)

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

SYNTAX:

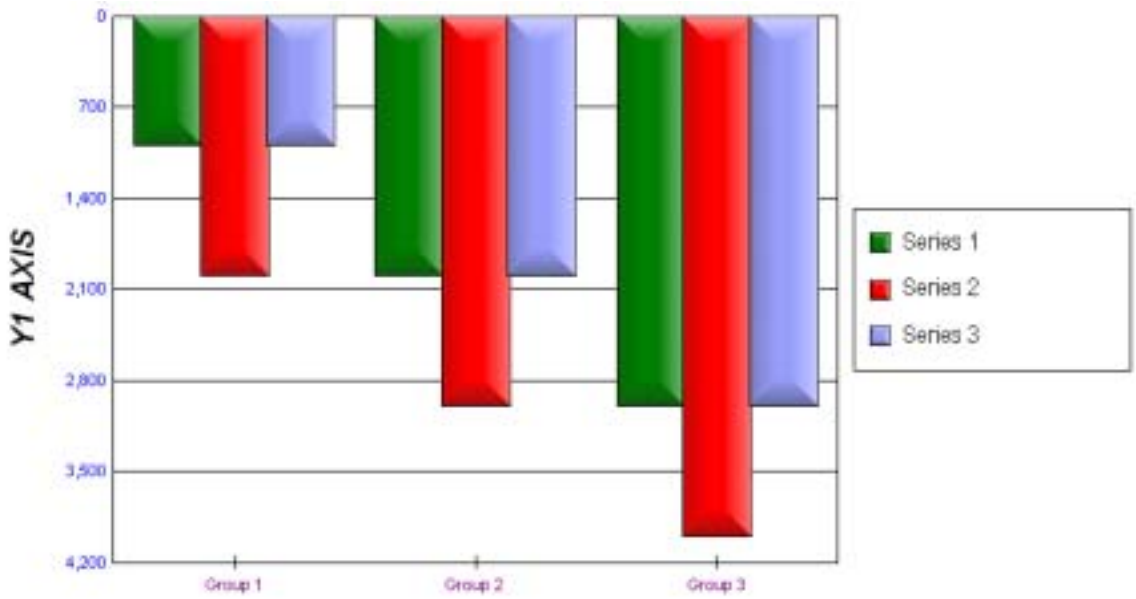
```
@Y1_INVERT bInvert
```

PARAMETERS:

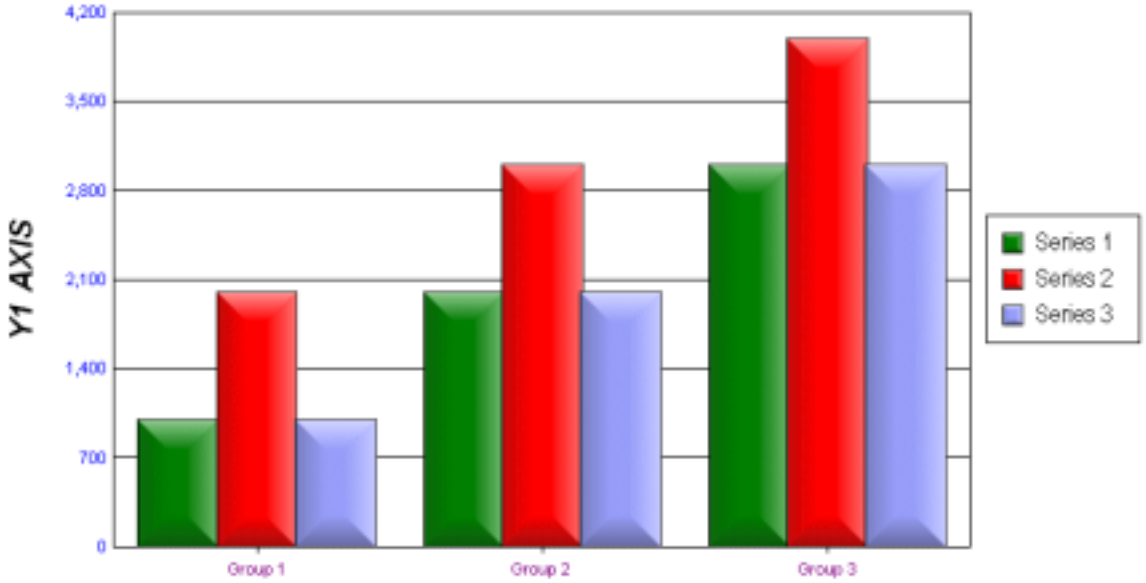
bInvert; 0=Draw values and labels normally. 1=Invert values and labels.

EXAMPLE:

```
@Y1_INVERT 1
```




```
@Y1_INVERT 0
```



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:

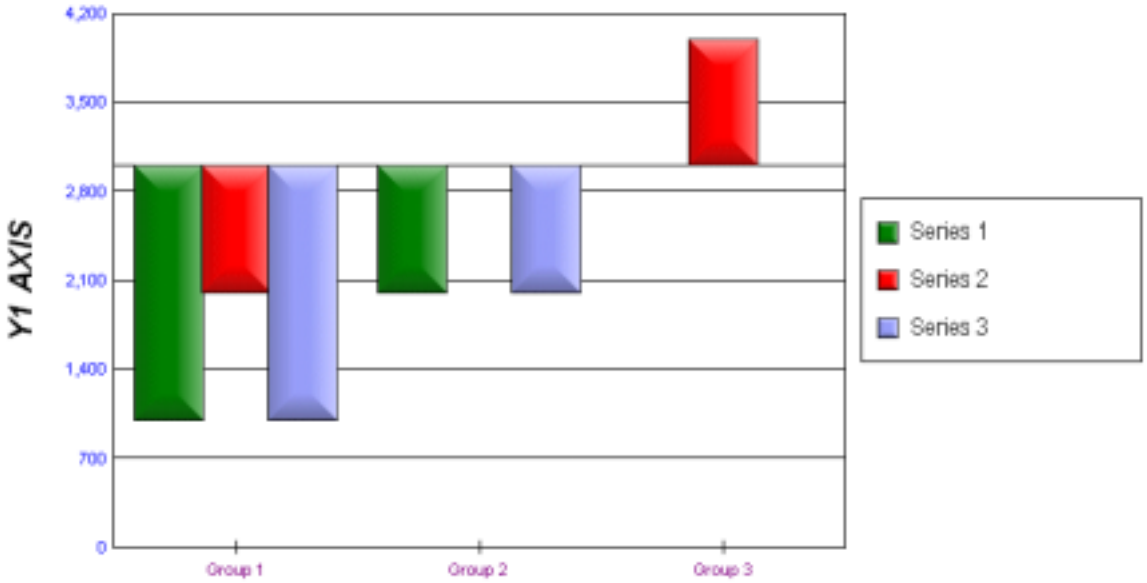
```
@Y1BASE fBase
```

PARAMETERS:

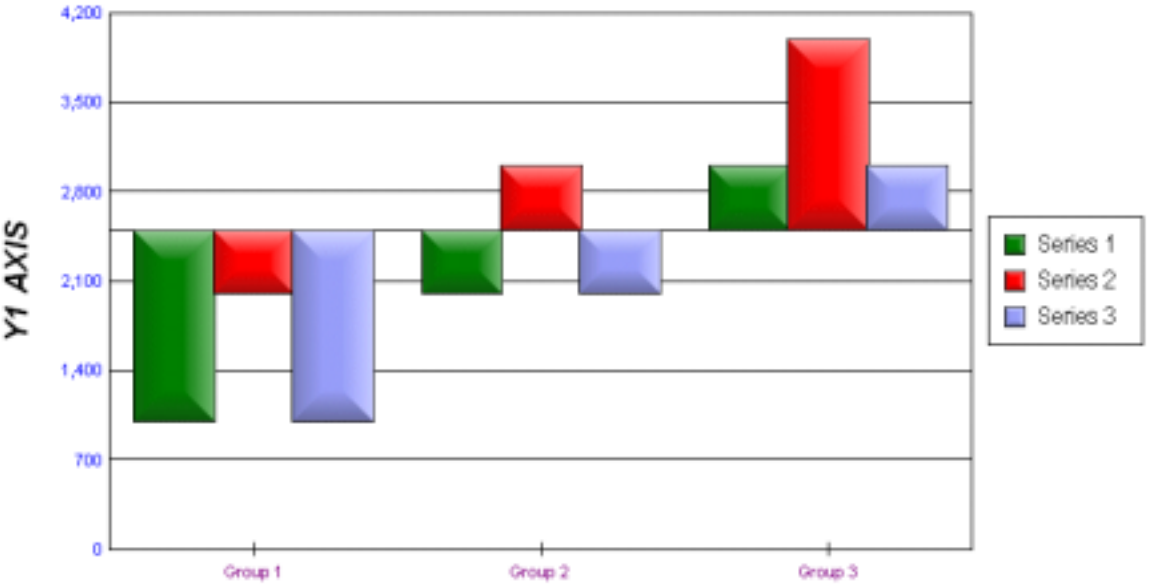
fBase; Base line value

EXAMPLE:

```
@Y1BASE 3000
```




```
@Y1BASE 2500
```



PERSISTENT:

NO

@Y2_FORCE_PERCENT (Y2-Axis Percent Format)

This macro can be used to reformat Y2-axis labels using percent format.

SYNTAX:

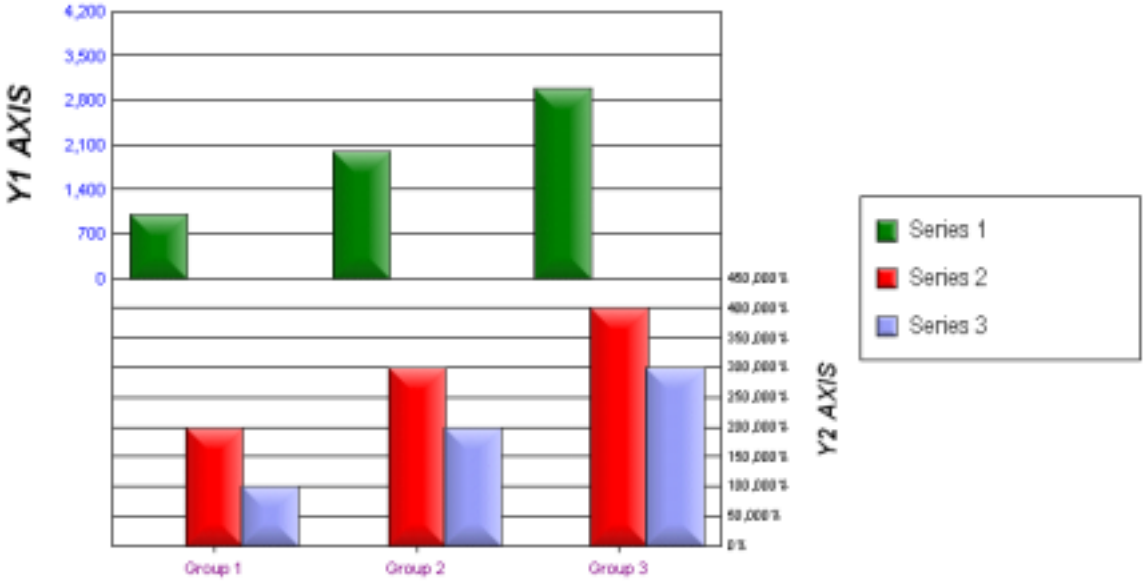
```
@Y2_FORCE_PERCENT bForce
```

PARAMETERS:

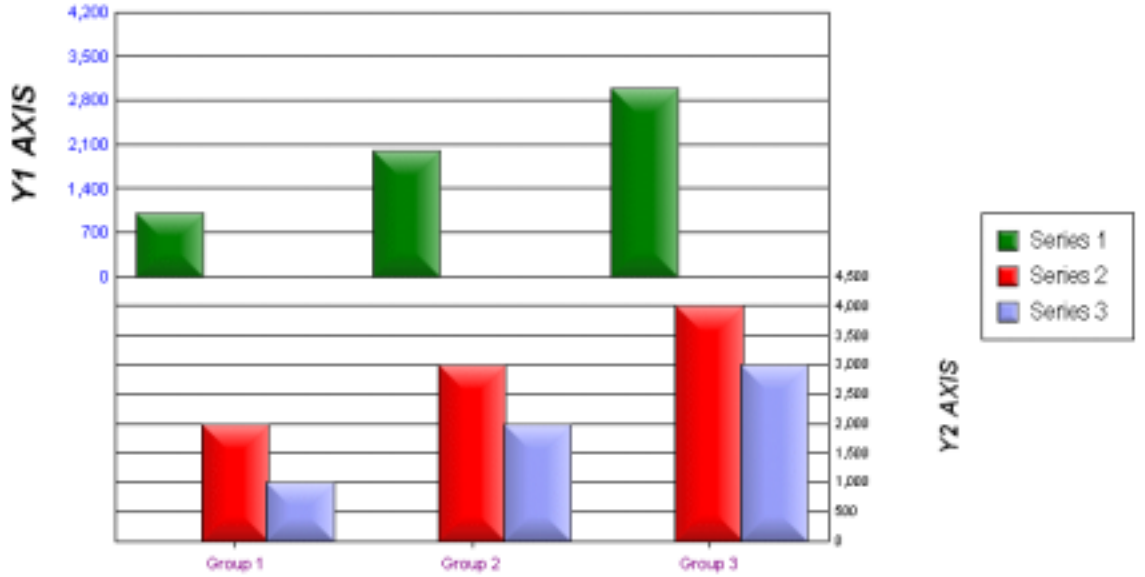
bForce; 0=Draw Y2-axis labels normally. 1=Use percent format.

EXAMPLE:

```
@Y2_FORCE_PERCENT 1
```




```
@Y2_FORCE_PERCENT 0
```



PERSISTENT:

NO

@Y2_INVERT (Y2-Axis Invert)

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

SYNTAX:

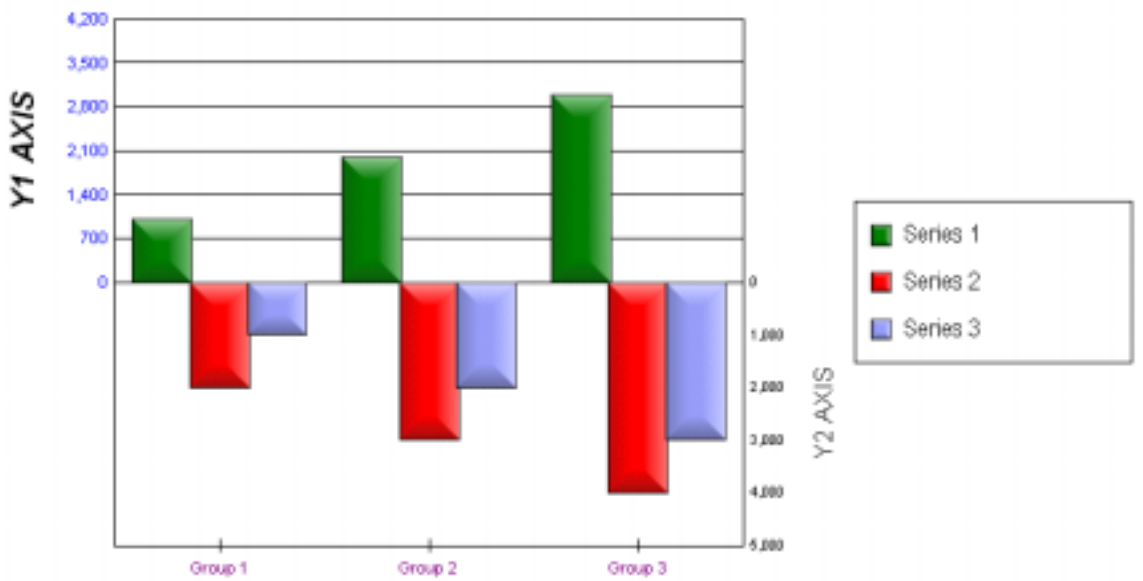
```
@Y2_INVERT bInvert
```

PARAMETERS:

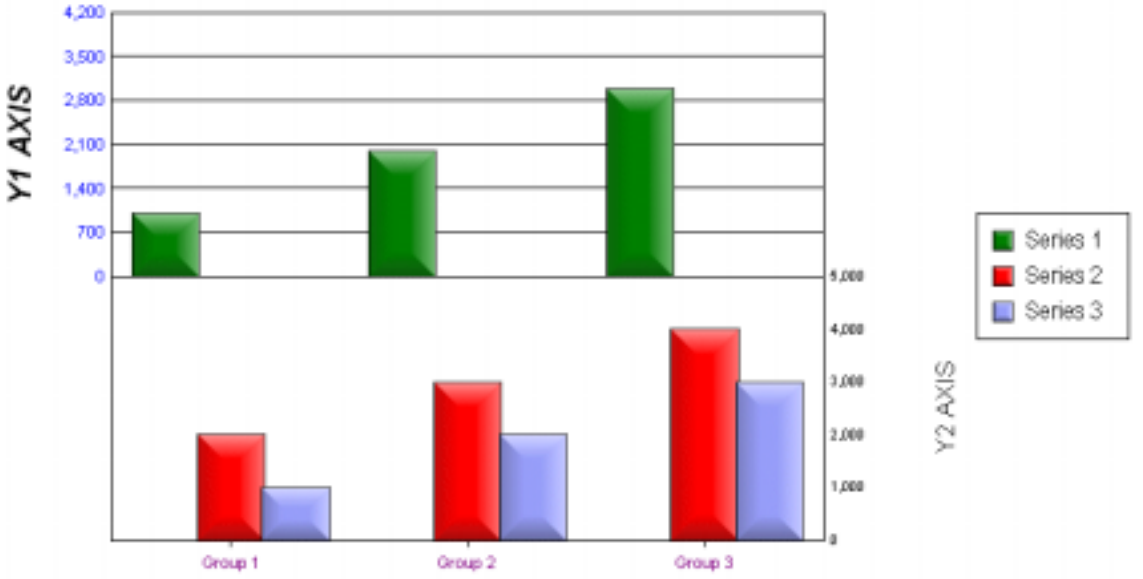
bInvert; 0=Draw values and labels normally. 1=Invert values and labels.

EXAMPLE:

```
@Y2_INVERT 1
```




```
@Y2_INVERT 0
```



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:

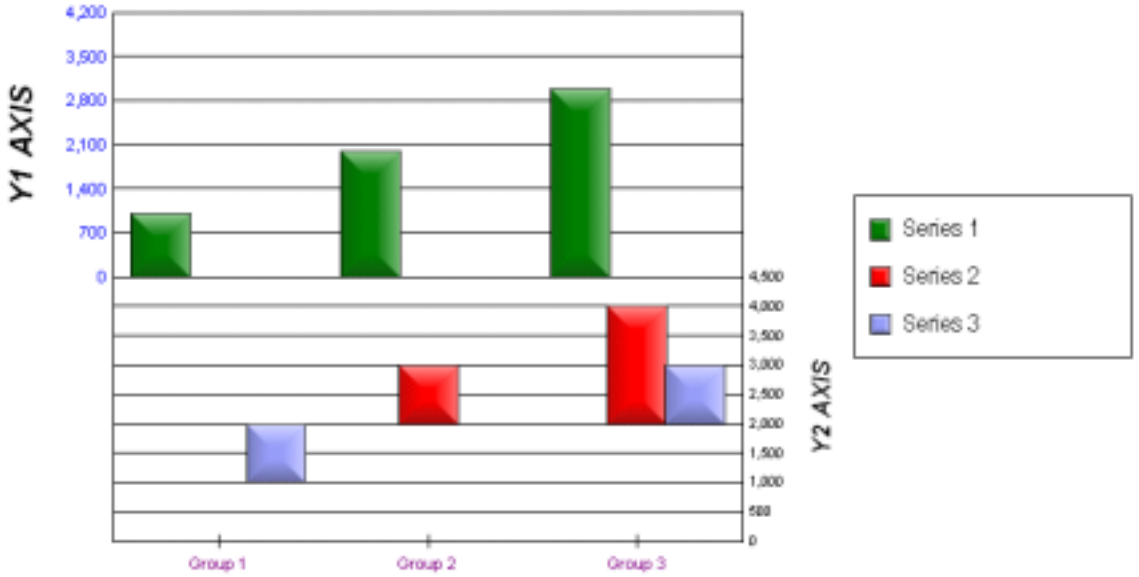
```
@Y2BASE fBase
```

PARAMETERS:

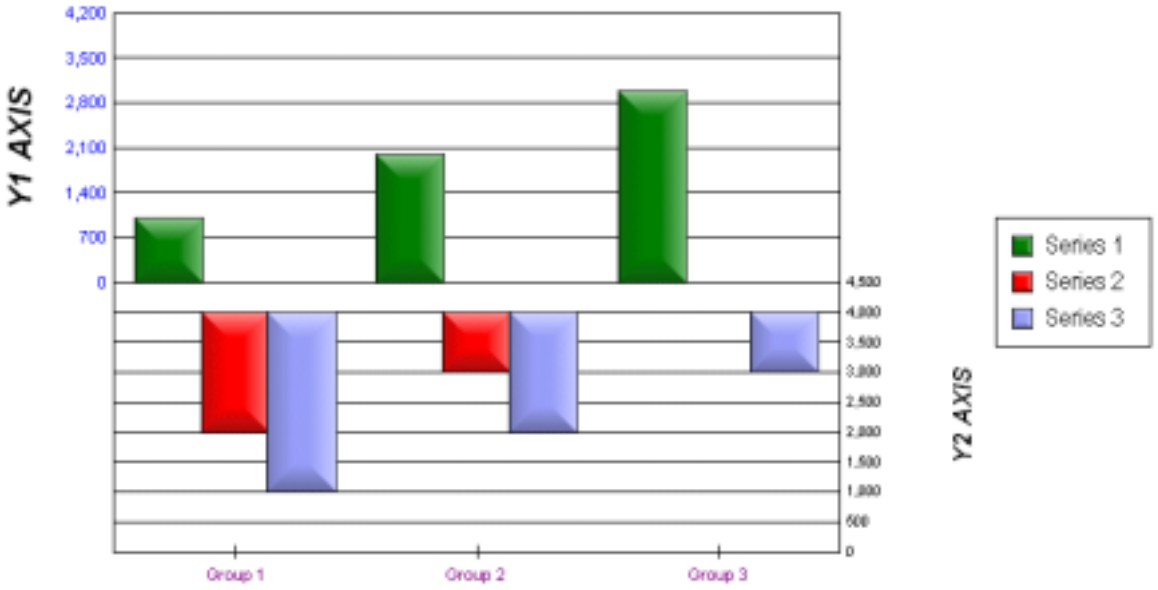
fBase; Base line value

EXAMPLE:

```
@Y2BASE 2000
```




```
@Y2BASE 4000
```



PERSISTENT:

NO

@Y2SLAVE (Y2-Axis Slave to Y1)

On a dual-Y axis chart, this macro forces the minimum/maximum values on the Y2-axis to be the same as the minimum/maximum values on the Y1-axis.

SYNTAX:

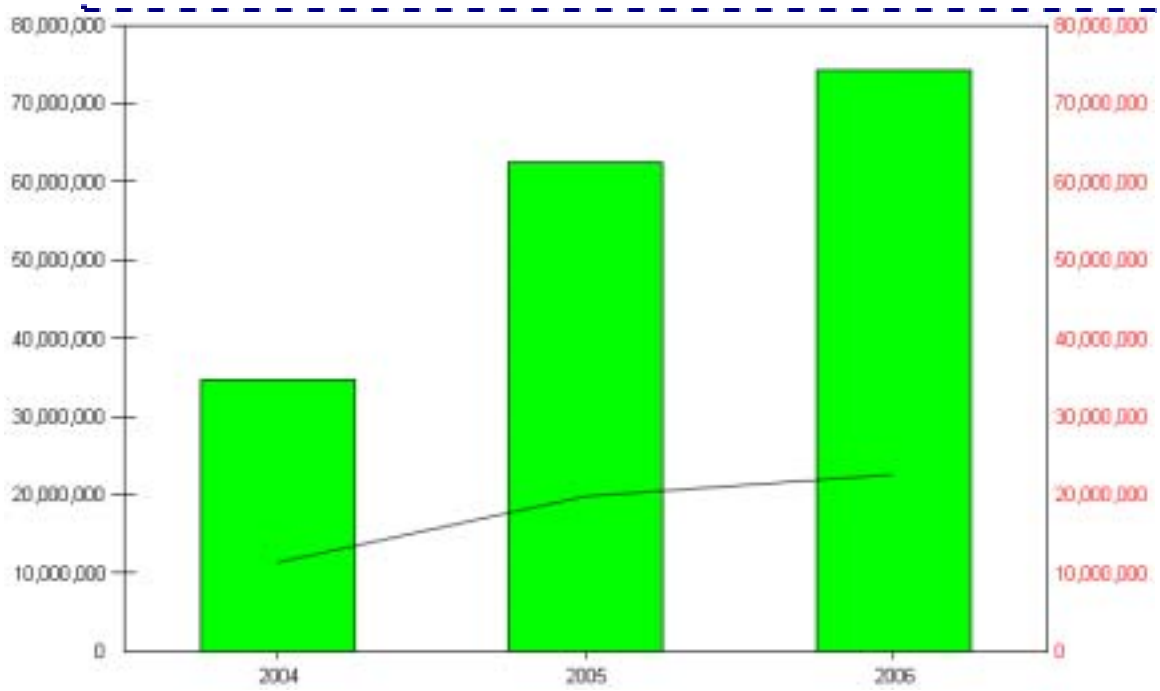
```
@Y2SLAVE
```

PARAMETERS:

None

EXAMPLE:

```
@Y2SLAVE
```



In this example, Y1-Axis values are drawn on the left side of the chart. Y2-Axis values are drawn on the right side of the chart.

PERSISTENT:

YES

ALSO SEE:

@Y2SLAVE2

@Y2SLAVE2 (Y1/Y2 Slave to Max Value)

On a dual-Y axis chart, this macro forces the minimum/maximum values to be the same on both axes (Y1 and Y2). ***Charts Unlimited*** determines the maximum value to use on both axes by calculating MAX (Y1's Maximum Value, Y2's Maximum Value) from the raw data that forms the chart.

SYNTAX:

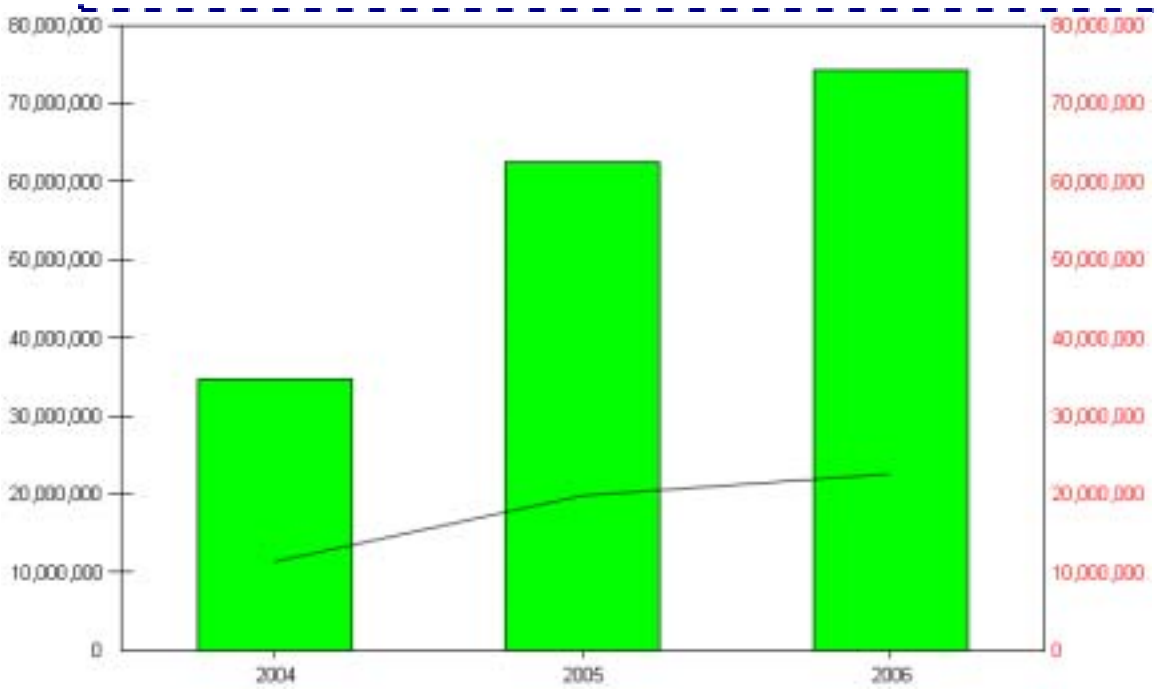
```
@Y2SLAVE2
```

PARAMETERS:

None

EXAMPLE:

```
@Y2SLAVE2
```



In this example, Y1-Axis values are drawn on the left side of the chart. Y2-Axis values are drawn on the right side of the chart.

PERSISTENT:

YES

ALSO SEE:

@Y2SLAVE

Section 4: Series & Groups

These macros can be used to control series and groups.

- @CALC_PERCENT_SERIES; Calculate Percentage Series
- @FORCE_ABSOLUTE; Force Series to plot Absolute
- @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
- @MIN_GROUPS; Add blank groups to control bar size
- @PERCENT_SERIES; Create a Ratio Series
- @RG; Reverse Groups
- @RS; Reverse Series
- @SINGLE_GROUP; Draw only the first Group in a Scatter Chart
- @SORT; Sort Series/Groups
- @STEP_LINE; Draw a Series Stepped Line
- @STEP_LINE2; Draw a Series Stepped Line at Values
- @SWAP; Swap Series/Groups
- @TOTAL_GROUP; Create a Total Group
- @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:

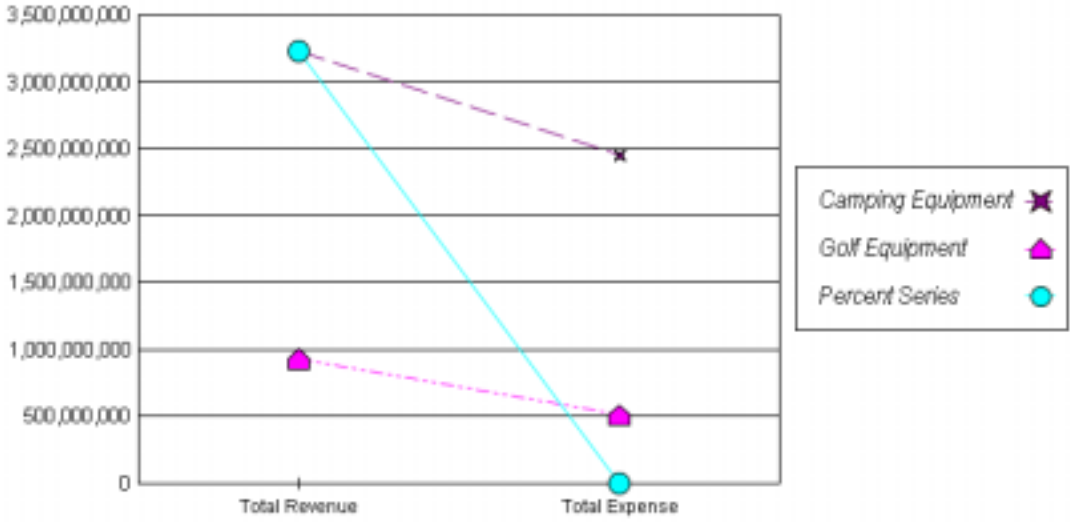
nNumeratorSeries; 0...n selects the numerator series

nDenominatorSeries; 0...n selects the denominator series

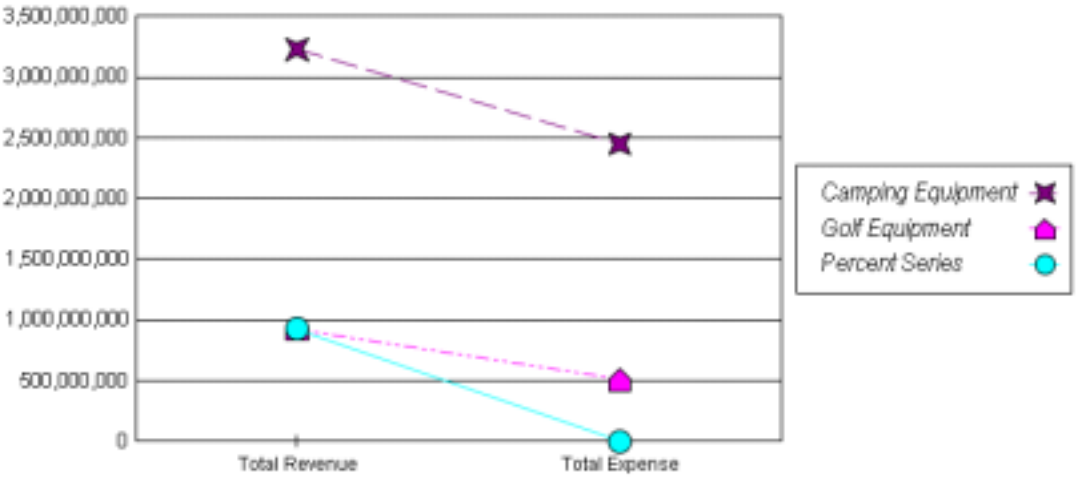
szSeriesLabel; new percent series label to use in legend area

EXAMPLE:

```
@CALC_PERCENT_SERIES 0 1 Percent Series
```




```
@CALC_PERCENT_SERIES 1 0 Percent Series
```



PERSISTENT:

NO

ALSO SEE:

@PERCENT_SERIES

@FORCE_ABSOLUTE (Force Series Absolute)

In any stacked chart, this macro forces a specified series (*nSeries*) to plot absolute.

SYNTAX:

```
@FORCE_ABSOLUTE nSeries bAbs
```

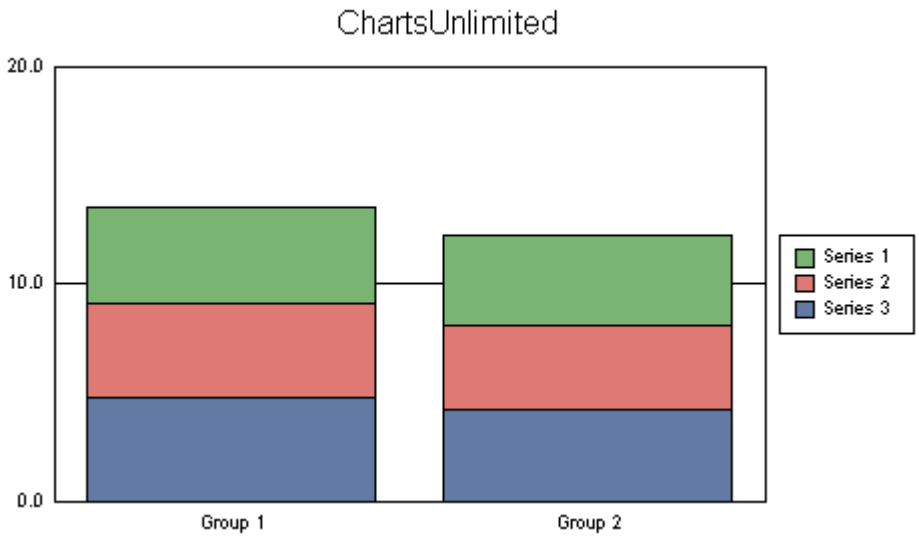
PARAMETERS:

nSeries; 0...999 defines the series to draw absolute (0=Series 1)

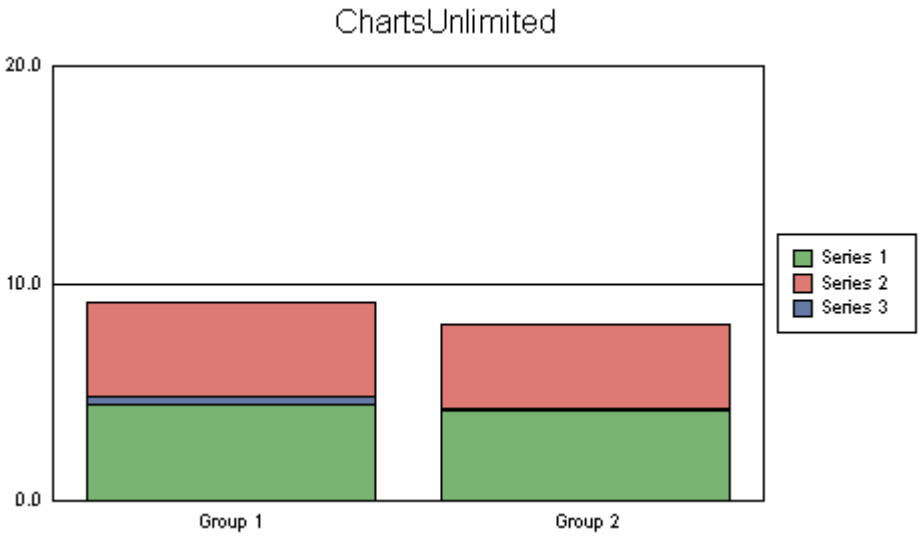
bAbs; 1 = Force absolute, 0= Keep Stacked

EXAMPLE:

```
@GRAPHTYPE 15 @FORCE_ABSOLUTE 2 0
```




```
@GRAPHTYPE 15 @FORCE_ABSOLUTE 2 1
```



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*. This macro is useful in some cases where Impromptu occasionally shows more data than the user is expecting.

SYNTAX:

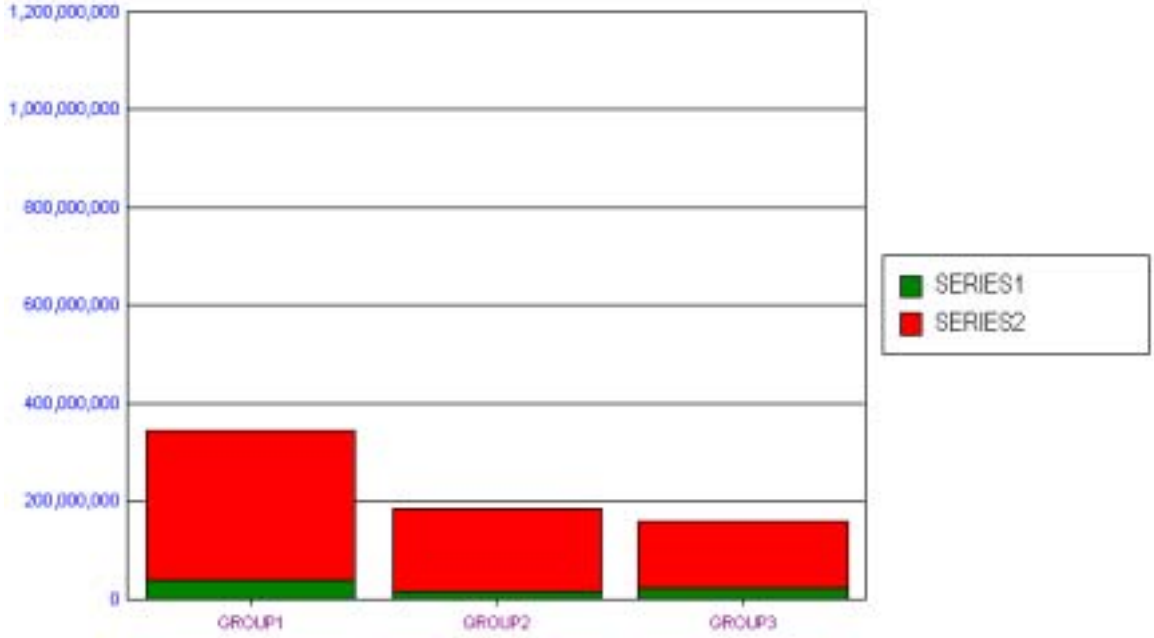
```
@FORCE_SERIES_COUNT nElements
```

PARAMETERS:

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

EXAMPLE:

```
@FORCE_SERIES_COUNT 2
```



PERSISTENT

NO

@FORECAST (Add Blank Groups)

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

SYNTAX:

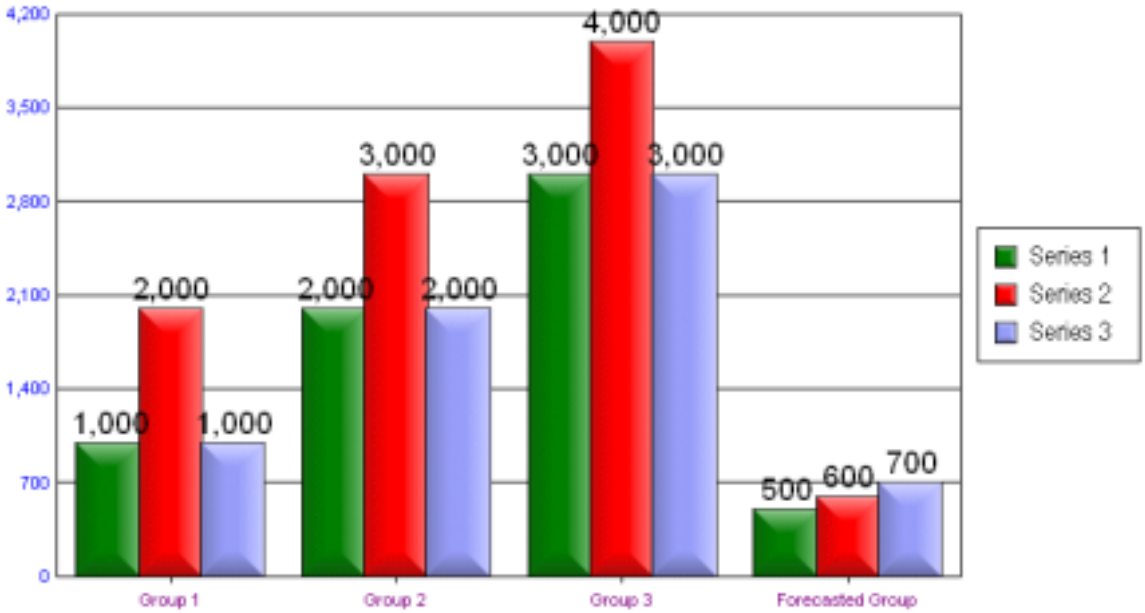
```
@FORECAST nPeriods
```

PARAMETERS:

nPeriods; 0…1024 groups

EXAMPLE:

```
@FORECAST 1
@DP 0 3 500
@DP 1 3 600
@DP 2 3 700
@AGL 3 Forecasted Group
@DATATEXT 1
```



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:

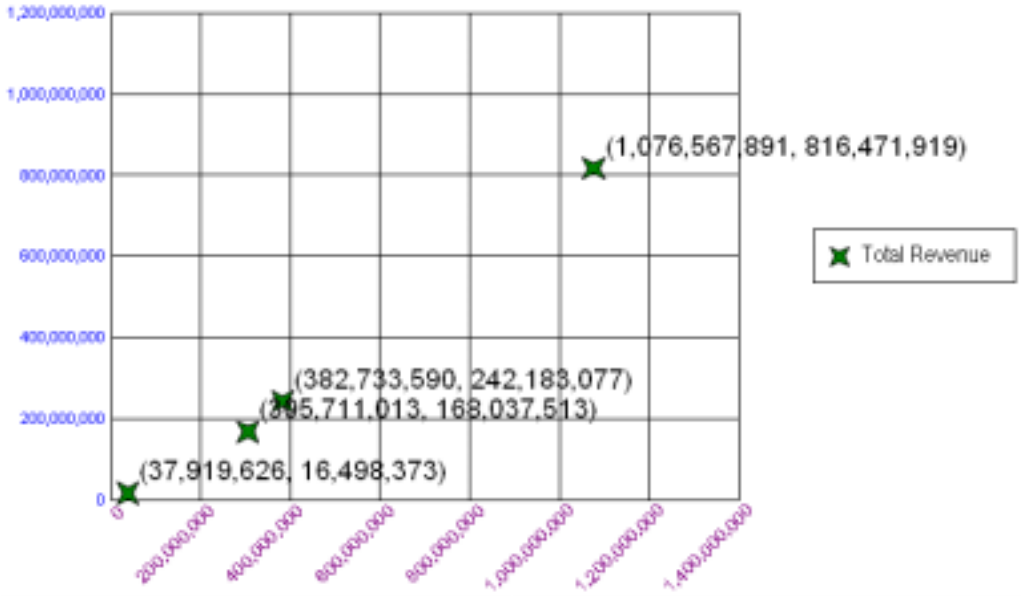
```
@GM bVertFormat
```

PARAMETERS:

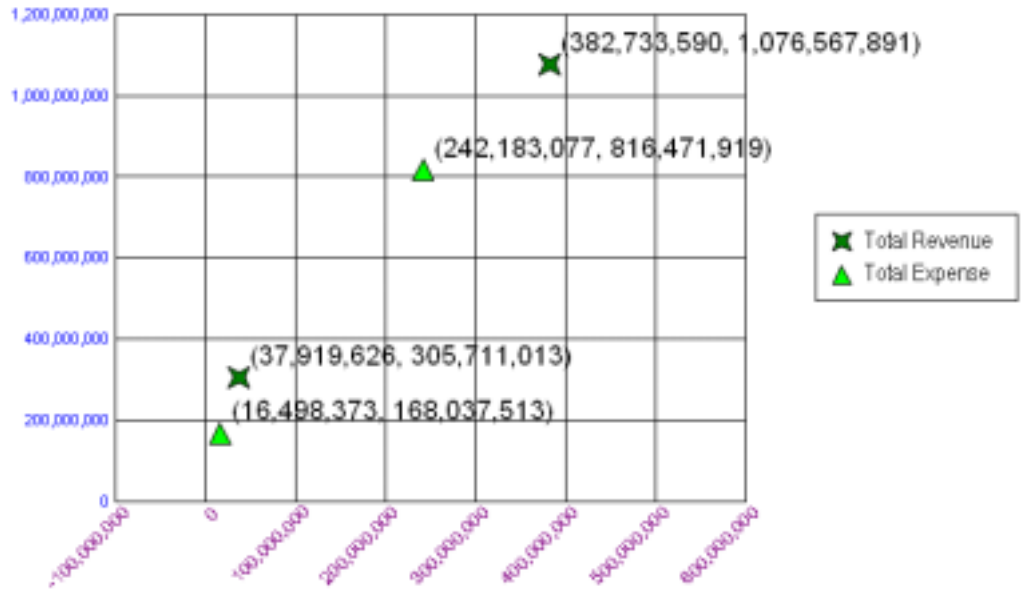
bVertFormat; 1 = true (read data in column major order), 0 = false (read data in row major order)

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:

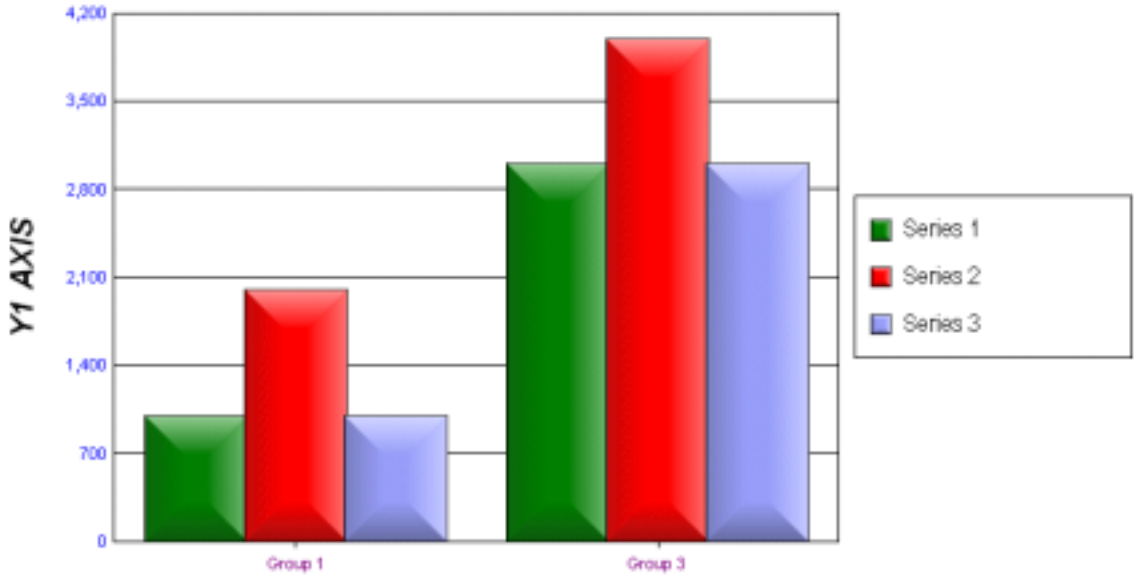
```
@IG nGroup
```

PARAMETERS:

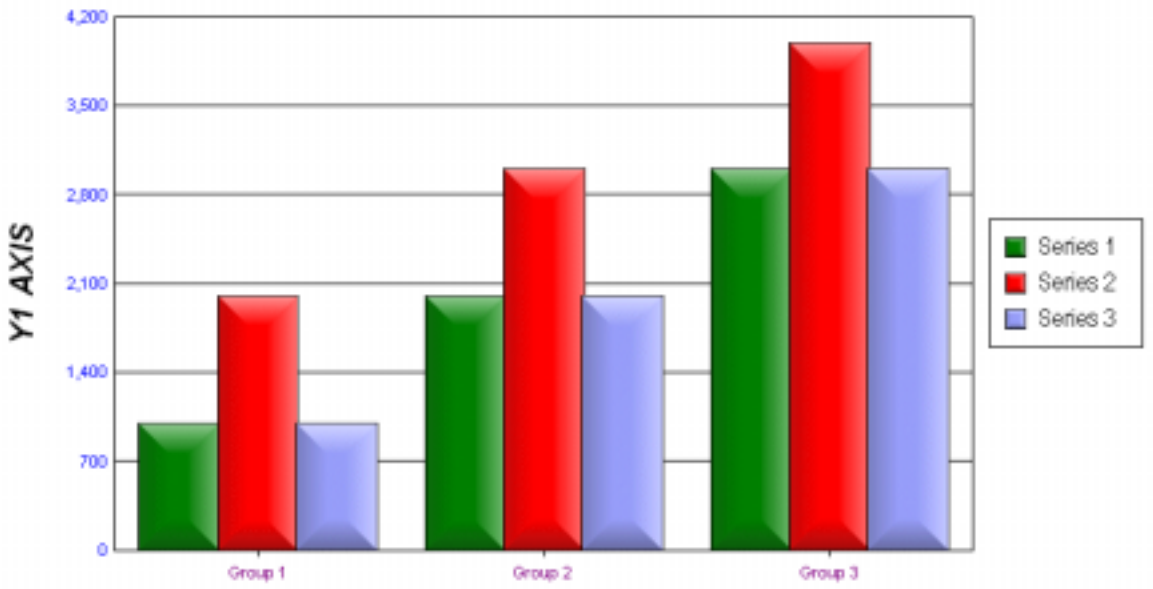
nGroup; Group (0...1024) to hide. -1=Restore all previously ignored groups.

EXAMPLE:

```
@IG 1
```




```
@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:

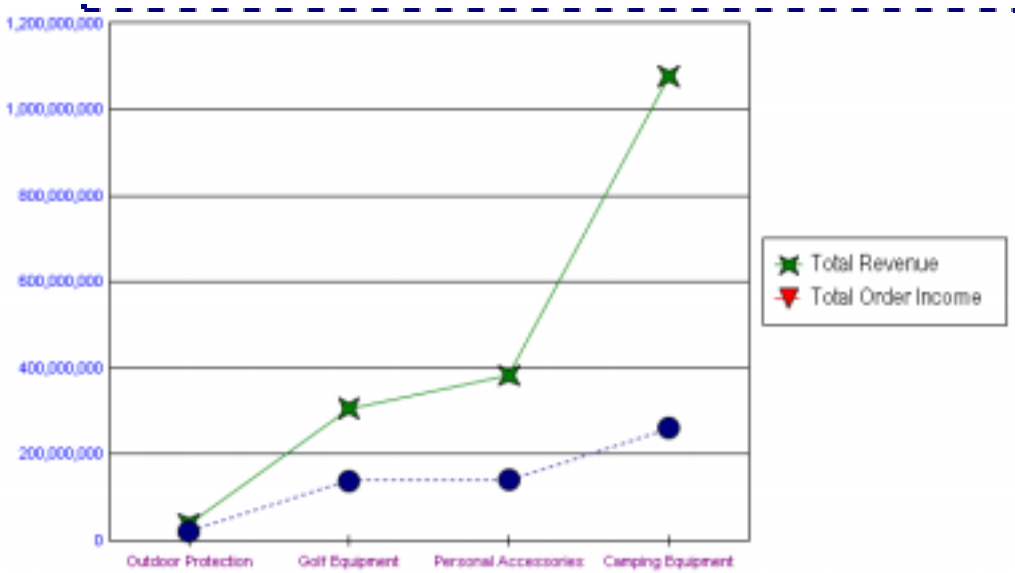
```
@IS nSeries
```

PARAMETERS:

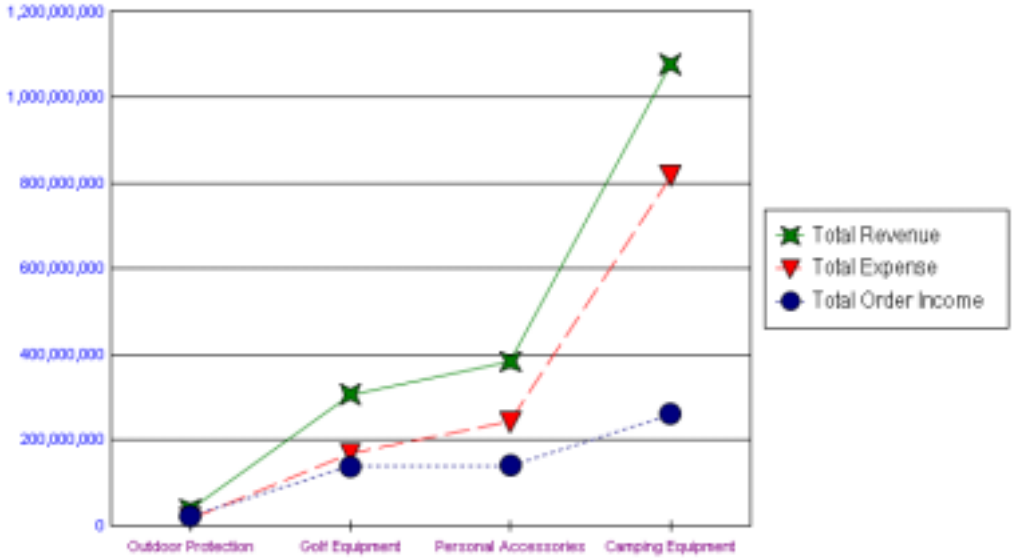
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1=restore all series that were previously ignored, 0=Series 1, 1=Series 2, etc.

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:

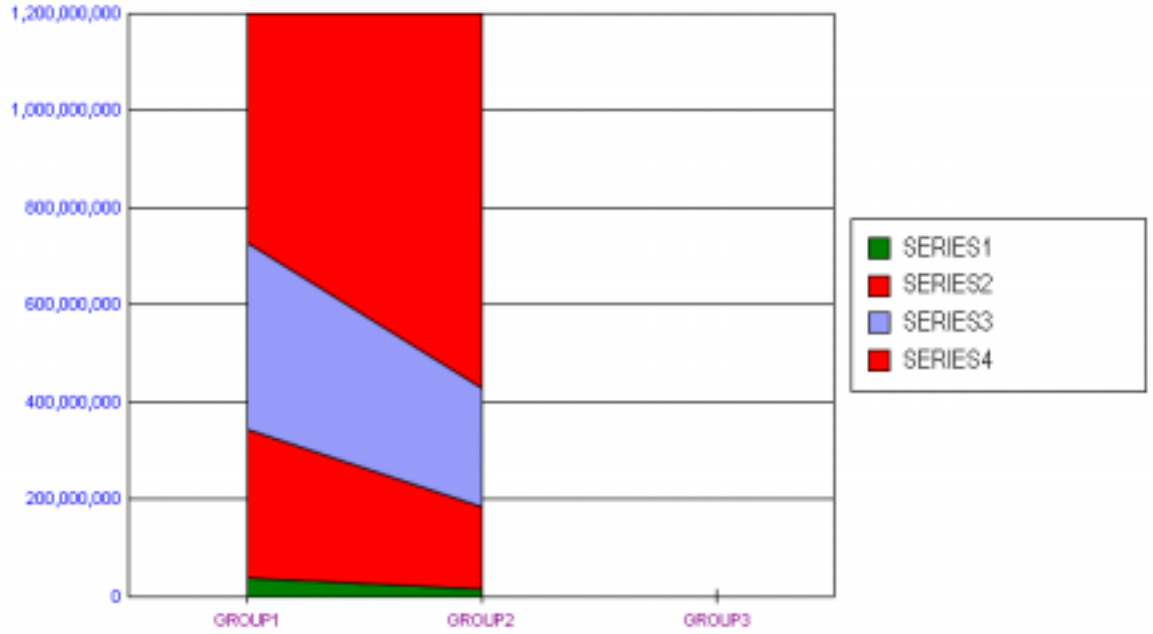
```
@LIMIT_VISIBLE_GROUPS nGroup
```

PARAMETERS:

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

EXAMPLE:

```
@LIMIT_VISIBLE_GROUPS 2
```



PERSISTENT:

YES

NOTES:

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

@GRAPHTYPE Value	Chart
1	Vertical Area Stacked
8	Horizontal Area Stacked

@MIN_GROUPS (Add Blank Groups/Control Bar Size)

This macro adds blank groups and NULL data so that a consistent size bar can be represented. So a chart with one bar and a chart with eight bars will draw the same size bar. The *nGroups* parameter specifies the static number of groups to use to calculate riser width (it is overwritten if there are more groups in data than *nGroups*).

SYNTAX:

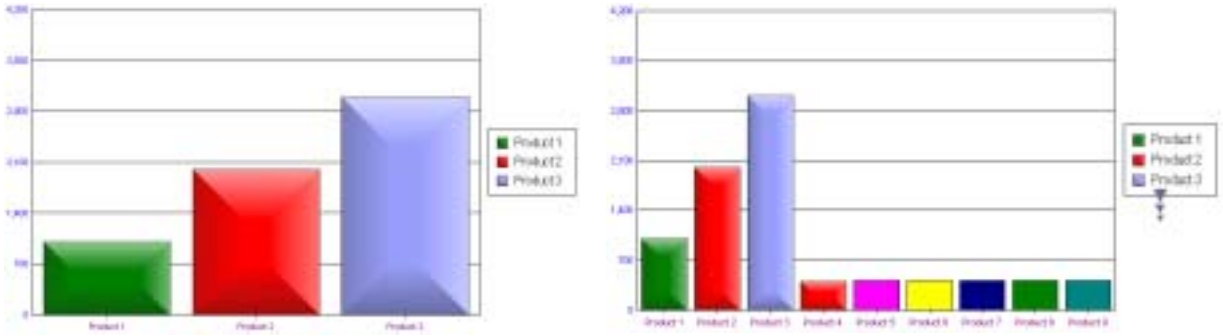
```
@MIN_GROUPS nGroups
```

PARAMETERS:

nGroups; static number of groups to pretend to use in the chart to calculate chart width

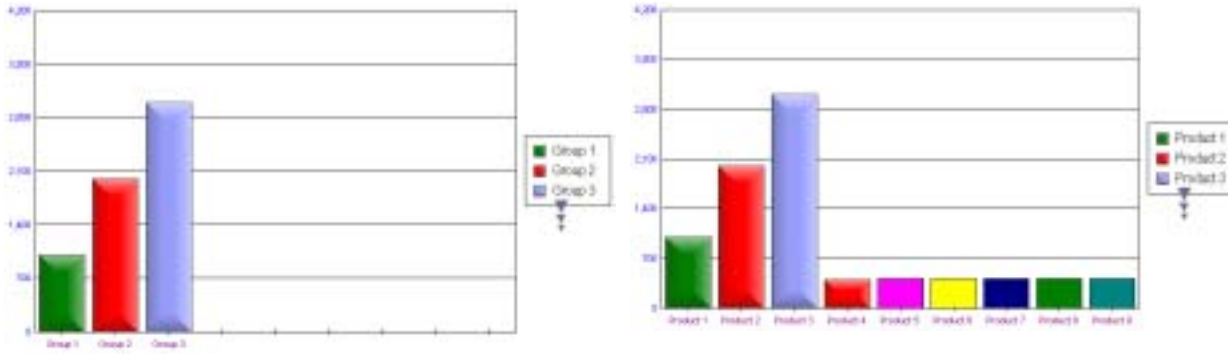
EXAMPLE:

This macro will GUARANTEE that the width of a bar in a single series chart will be identical regardless of the number groups. For example, a multi-page report where each page represents the sales of a product in a given state may have between 1 and 9 groups. Some states only carry one product, some carry 3 products, some carry ALL products. Without this macro, some states will have very wide bars:



@MIN_GROUPS 9 insures an eye-pleasing set of charts with consistent bar sizes.

```
@MIN_GROUPS 9
```



PERSISTENT:

NO

@PERCENT_SERIES (Create Ratio Series)

This macro can be used to change the values of an existing series in a chart to show the ratio of two other series. The value of the ratio series (at *nTarget*) is calculated by dividing the value of series *nTop* by the value of the series at *nBottom*. Note that the target series (*nTarget*) must already exist in the chart.

SYNTAX:

```
@PERCENT_SERIES nTop nBottom nTarget
```

PARAMETERS:

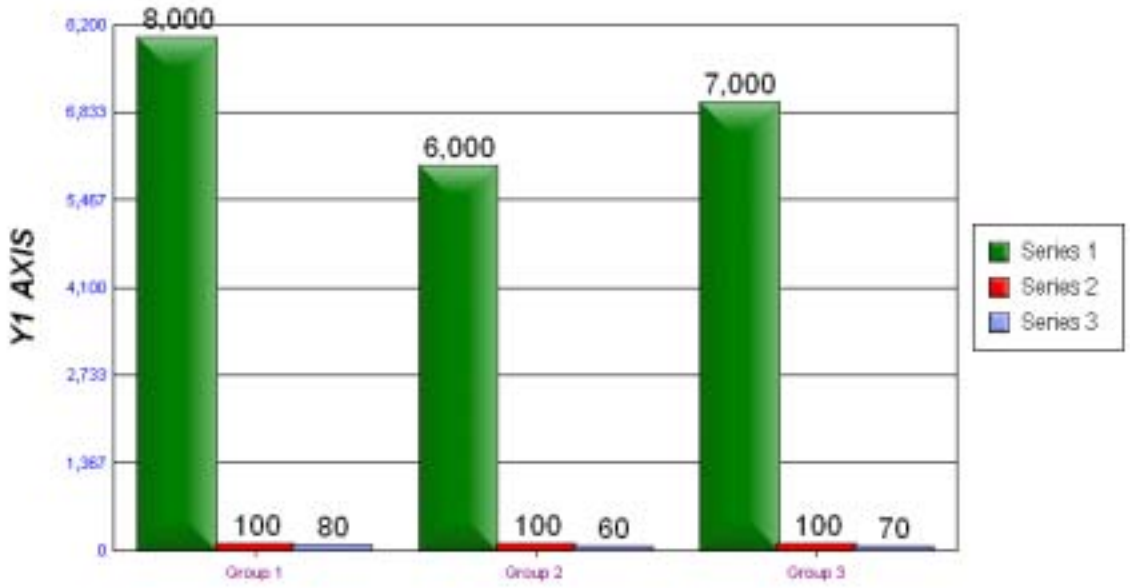
nTop; 1...*n* (where: *n* = the total number of series in the chart) selects the first source series.

nBottom; 1...*n* (where: *n* = the total number of series in the chart) selects the second source series.

nTarget; 1...*n* selects the target series. It must be a series that currently exists in the chart.

EXAMPLE:

```
@PERCENT_SERIES 0 1 2
```



PERSISTENT:

NO

@RG (Reverse Group)

This macro reverses the order of groups in a chart.

SYNTAX:

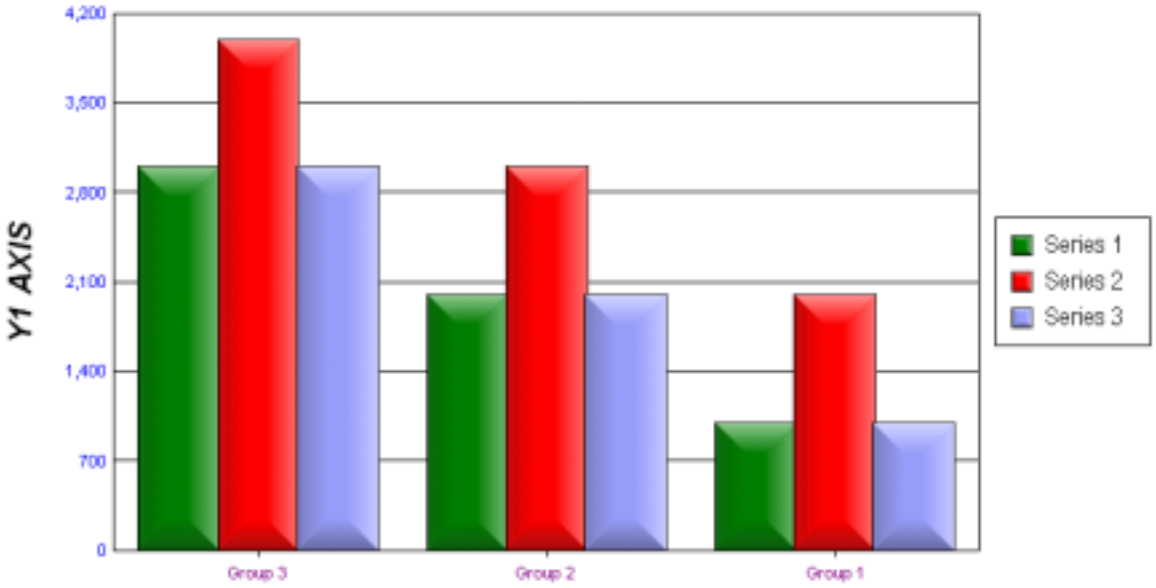
```
@RG bColReverse
```

PARAMETERS:

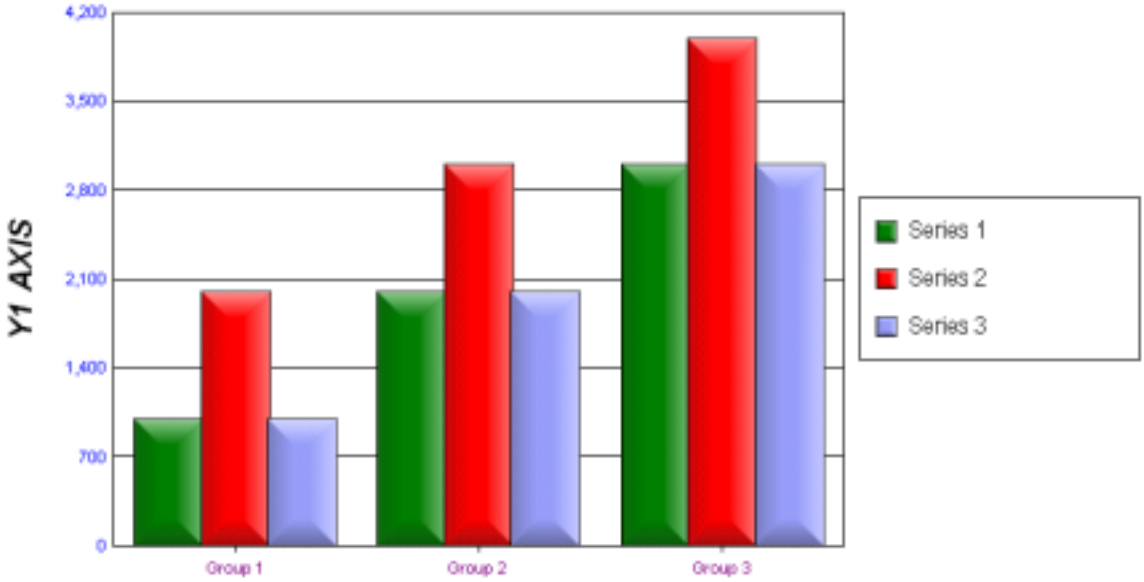
bColReverse; 1=true/reverse groups, 0=false/do not reverse groups.

EXAMPLE:

```
@RG 1
```




```
@RG 0
```



PERSISTENT:

YES

ALSO SEE:

@RS

@RS (Reverse Series)

This macro reverses the order of series in a chart.

SYNTAX:

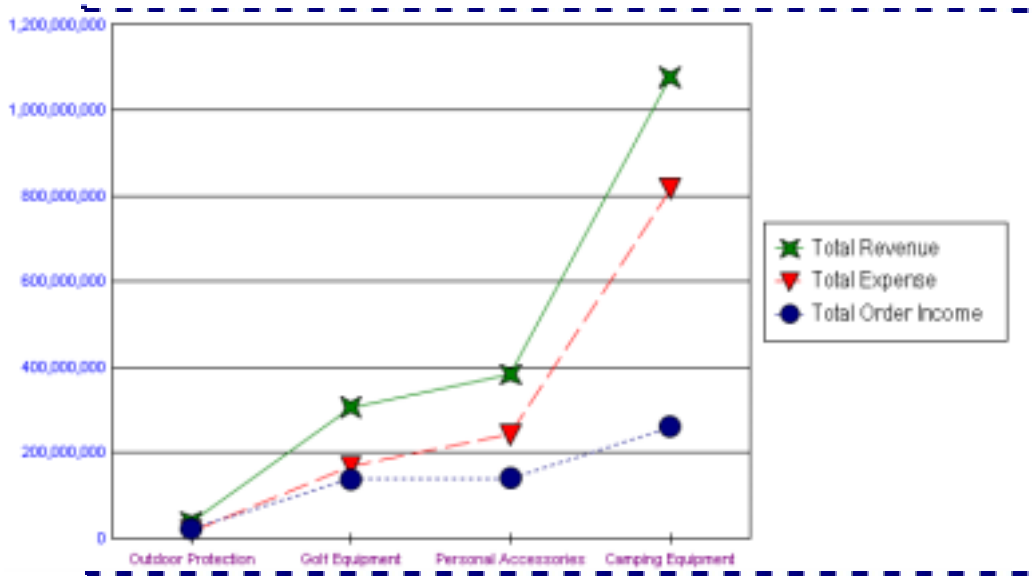
```
@RS bRowReverse
```

PARAMETERS:

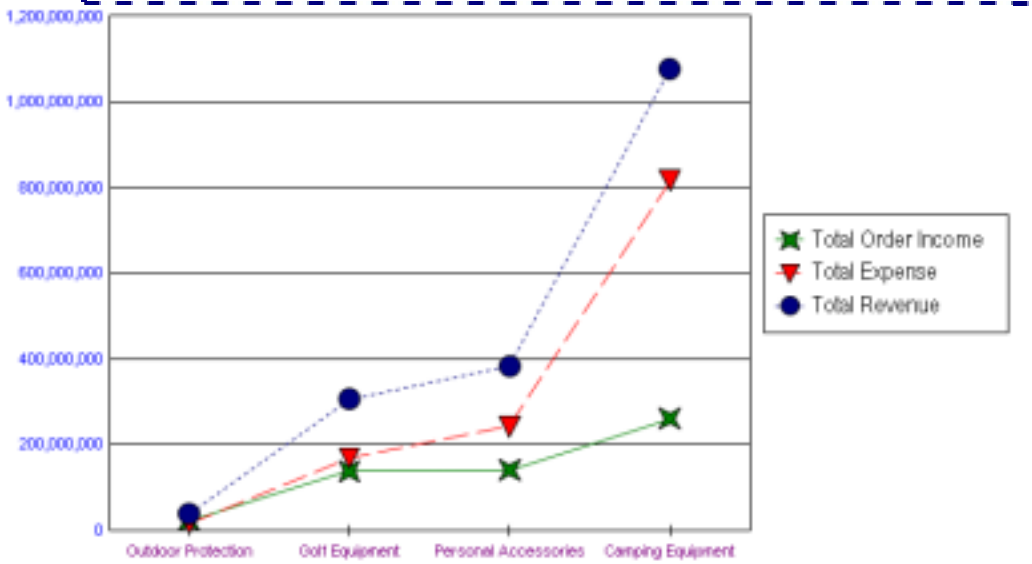
bRowReverse; 1=true/reverse series, 0=false/do not reverse series.

EXAMPLE:

```
@RS 0
```




```
@RS 1
```



PERSISTENT:

YES

ALSO SEE:

@RG

@SINGLE_GROUP (Single Group in Scatter Chart)

For scatter charts, this macro forces the chart to only display the first group.

SYNTAX:

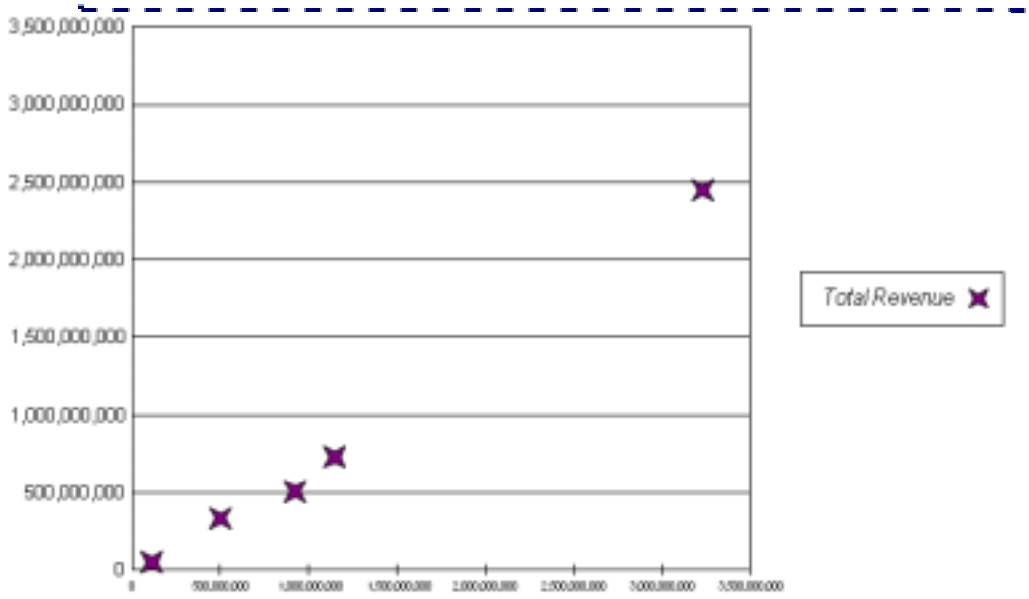
```
@SINGLE_GROUP
```

PARAMETERS:

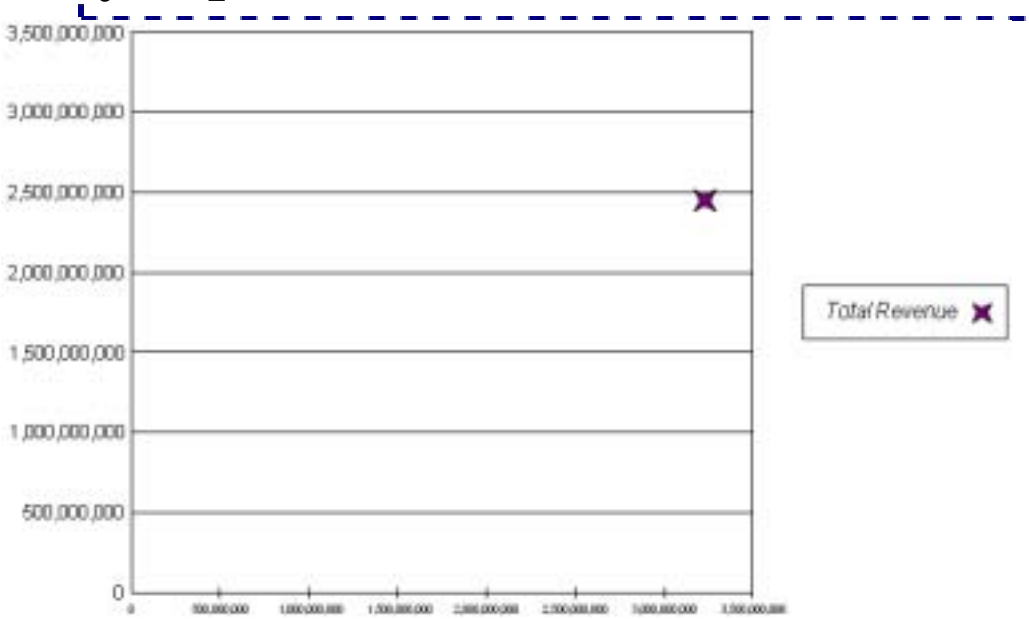
None

EXAMPLE:

```
@GRAPHTYPE 50
```




```
@SINGLE_GROUP
```



PERSISTENT:

NO

@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:

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

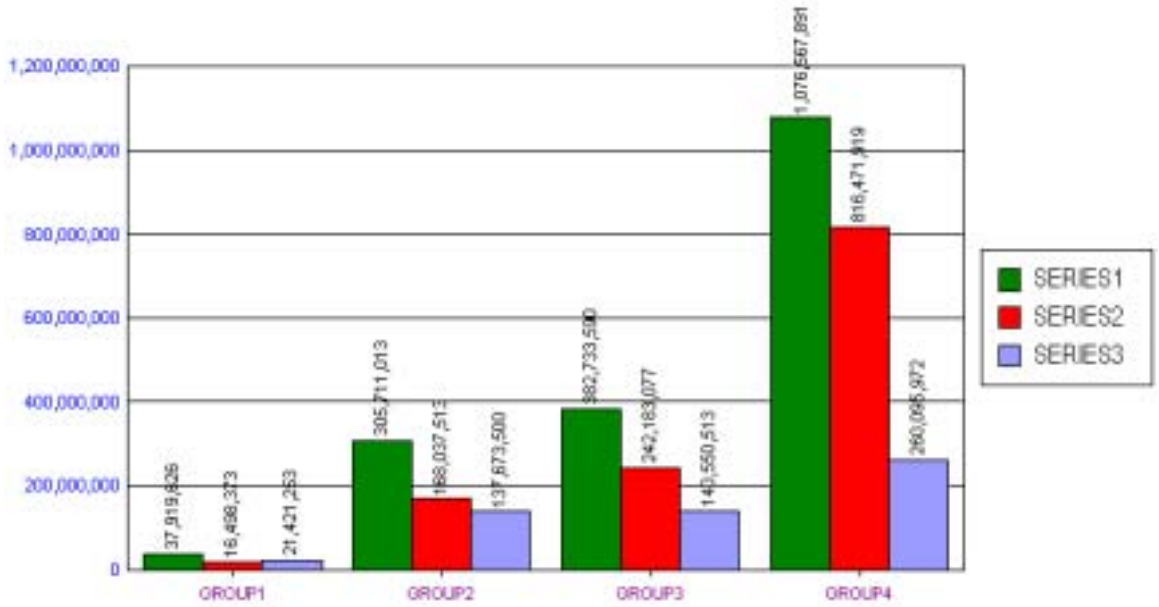
7 = Sort Groups Totals in descending numeric value

8 = First Series Key to Groups Ascending

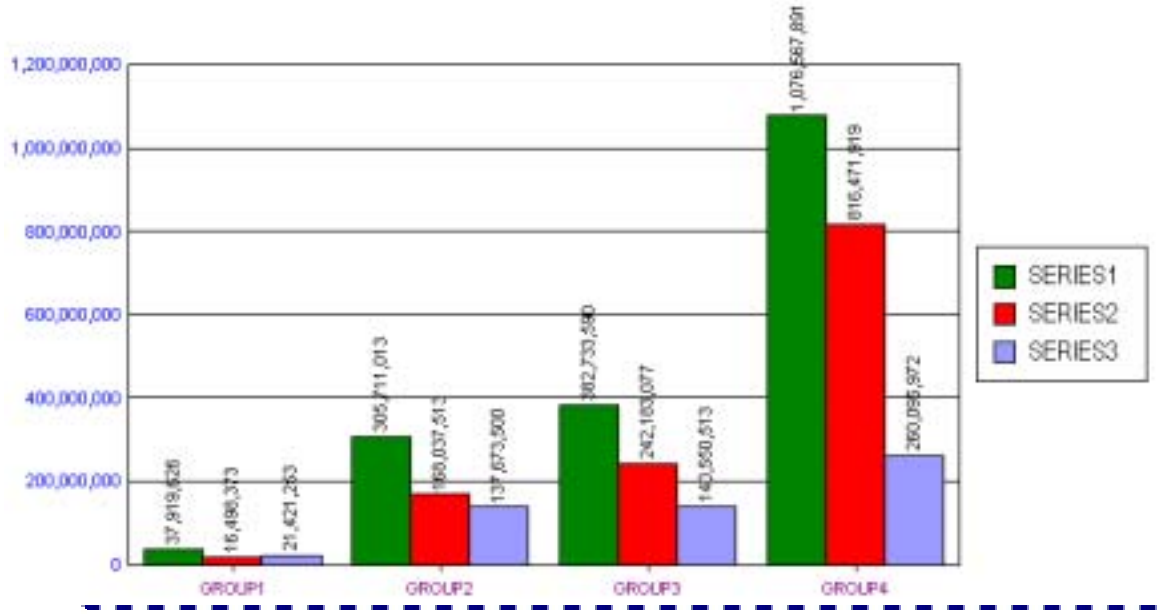
9 = First Series Key to Groups Descending

EXAMPLE:

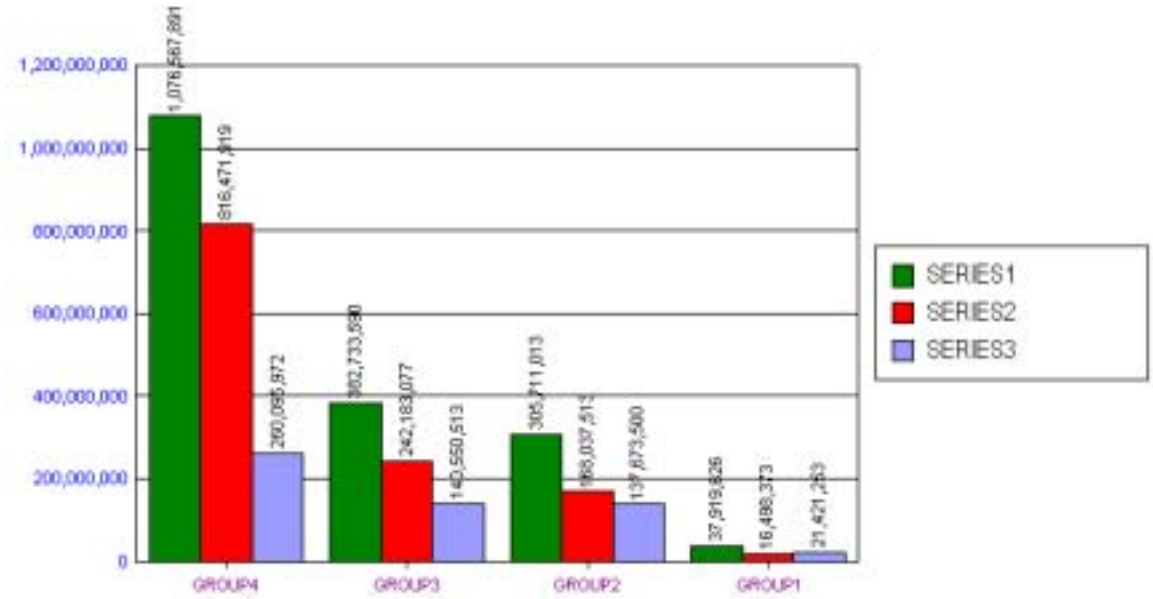
```
@DATATEXT 1 @RDT 3 @SORT 1
```




```
@SORT 2
```



@SORT 3



PERSISTENT:

NO

@STEP_LINE (Series Stepped Line)

In a line chart, this macro can be used to change any or all series in the chart to draw as a stepped line.

SYNTAX:

```
@STEP_LINE nSeries bStep
```

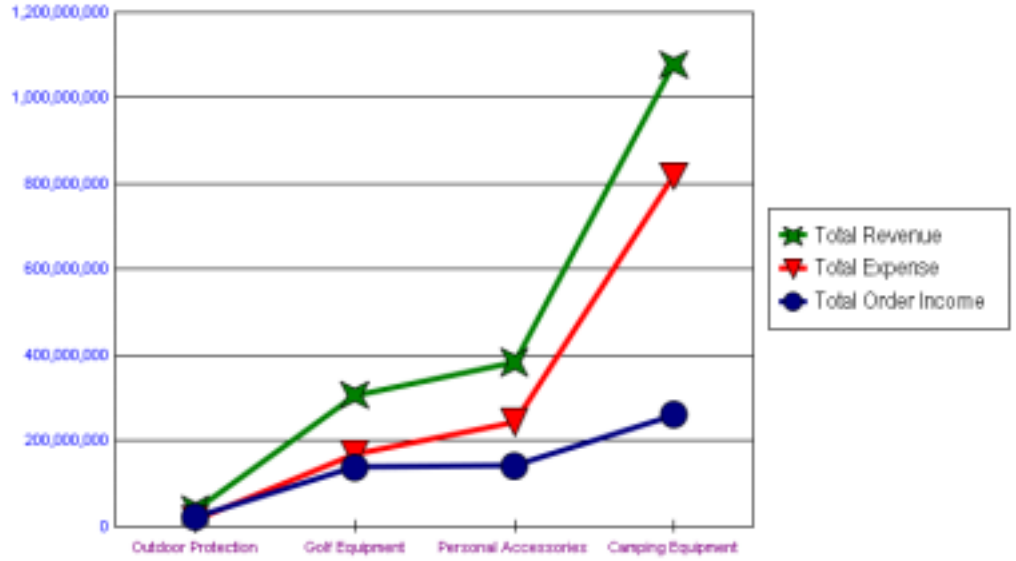
PARAMETERS:

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.

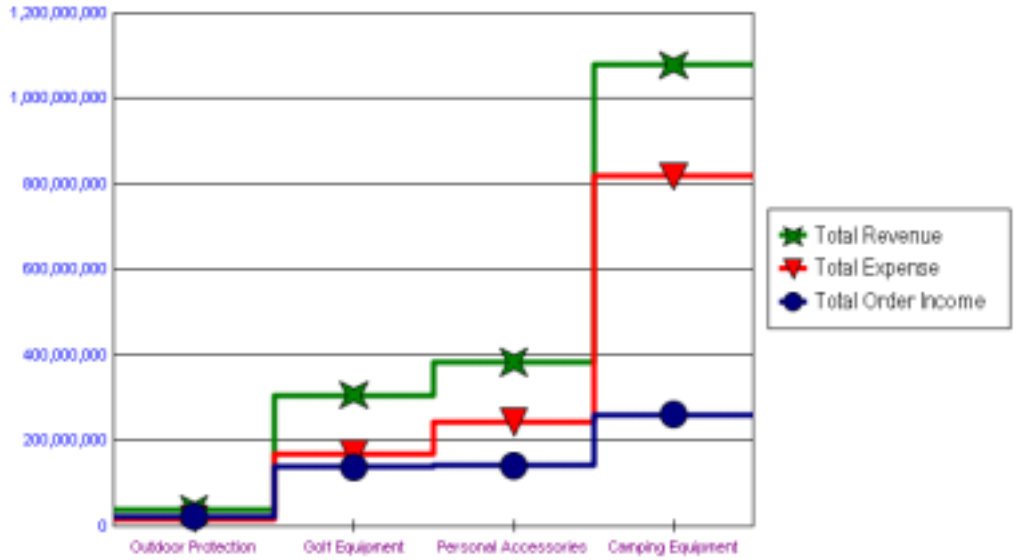
bStep; 1 = draw *nSeries* as stepped line. 0 = draw *nSeries* normally.

EXAMPLE:

```
@SZ 50 @LS -1 300 0 @STEP_LINE -1 0
```




```
@SZ 50 @LS -1 300 0 @STEP_LINE -1 1
```



PERSISTENT:

YES

@STEP_LINE2 (Series Stepped Line at Values)

In a line chart, this macro can be used to change any or all series in the chart to draw as a stepped line that steps at values (not between).

SYNTAX:

```
@STEP_LINE2 nSeries bStep
```

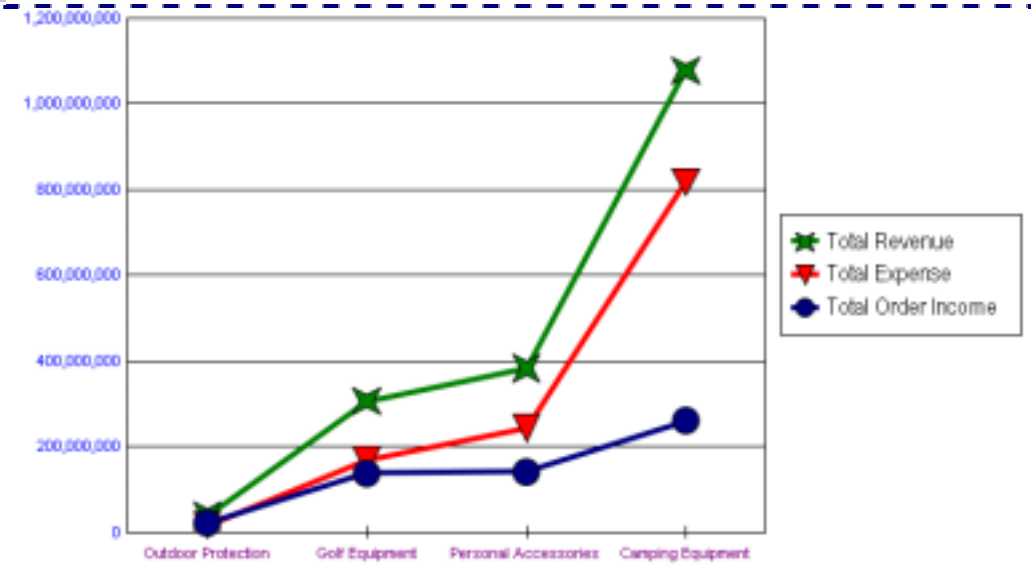
PARAMETERS:

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.

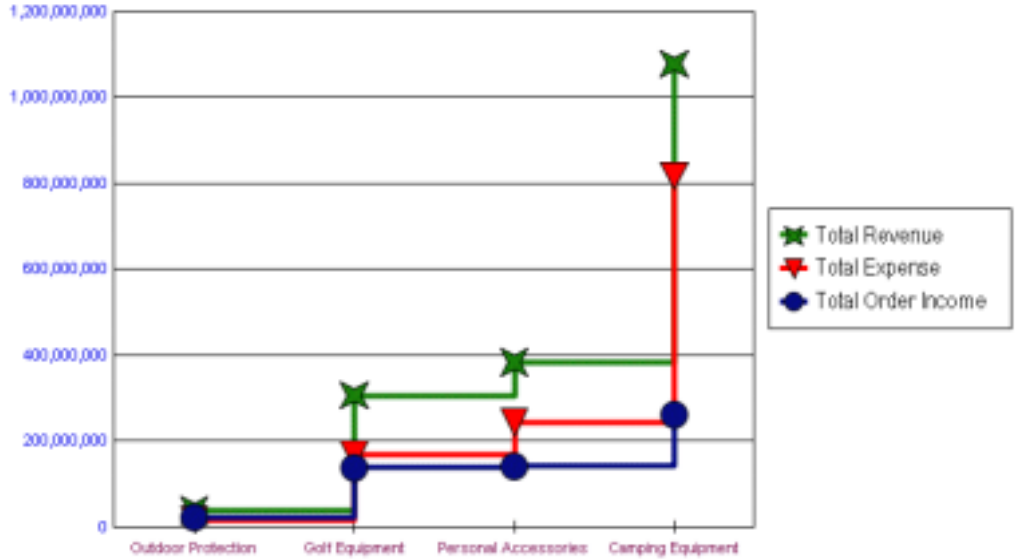
bStep; 1 = draw *nSeries* as stepped line. 0 = draw *nSeries* normally.

EXAMPLE:

```
@SZ 50 @LS 1 300 0 @STEP_LINE2 -1 0
```




```
@SZ 50 @LS 1 300 0 @STEP_LINE2 -1 1
```



PERSISTENT:

YES

@SWAP (Swap Series/Groups)

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

SYNTAX:

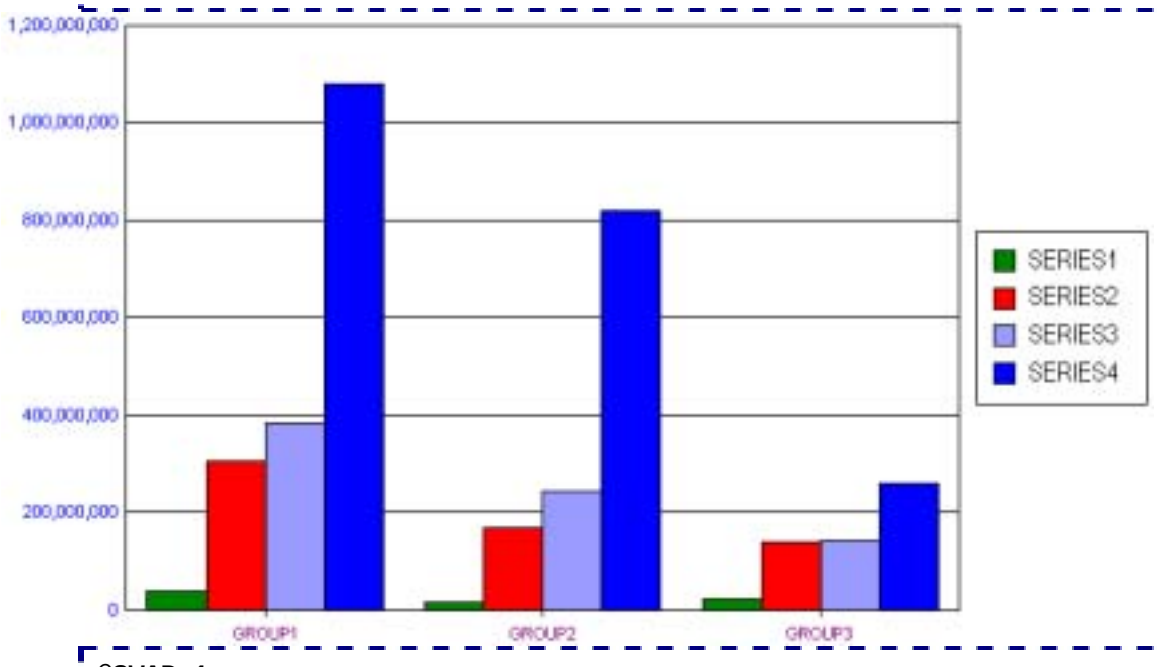
```
@SWAP bSwap
```

PARAMETERS:

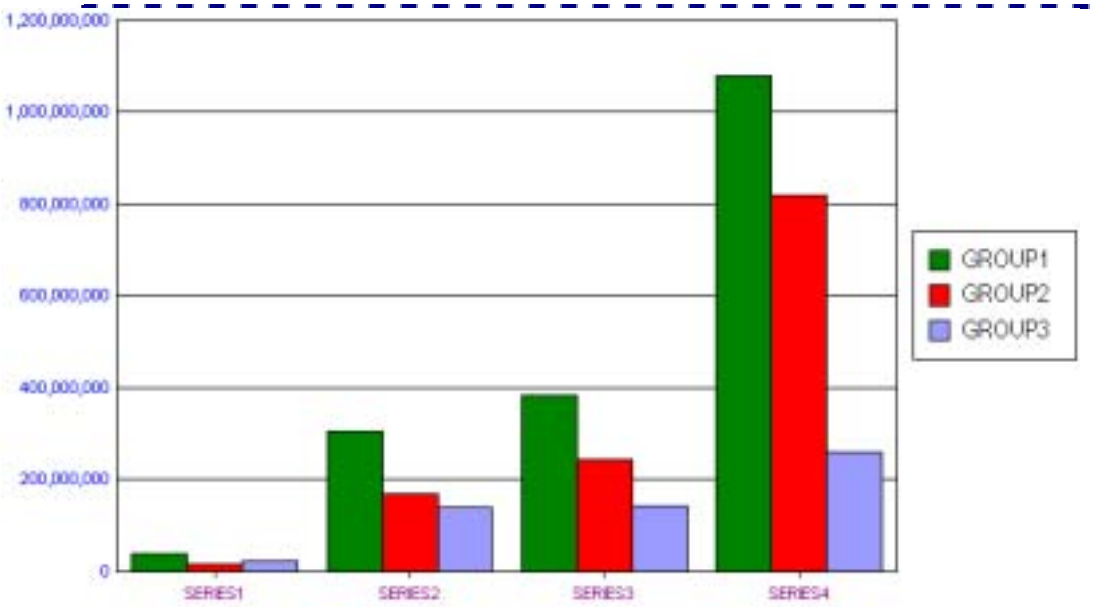
bSwap; 1 = swap series/group orientation, 0= do not swap series/group orientation

EXAMPLE:

```
@SWAP 0
```




```
@SWAP 1
```



PERSISTENT:

YES

@TOTAL_GROUP (Create a Total Group)

This macro adds a new group to a chart and populates it with risers that are equal to the total of all previous groups in the chart. The optional label parameter (*szLabel*) can be used to label the group on the Group/O1-axis.

SYNTAX:

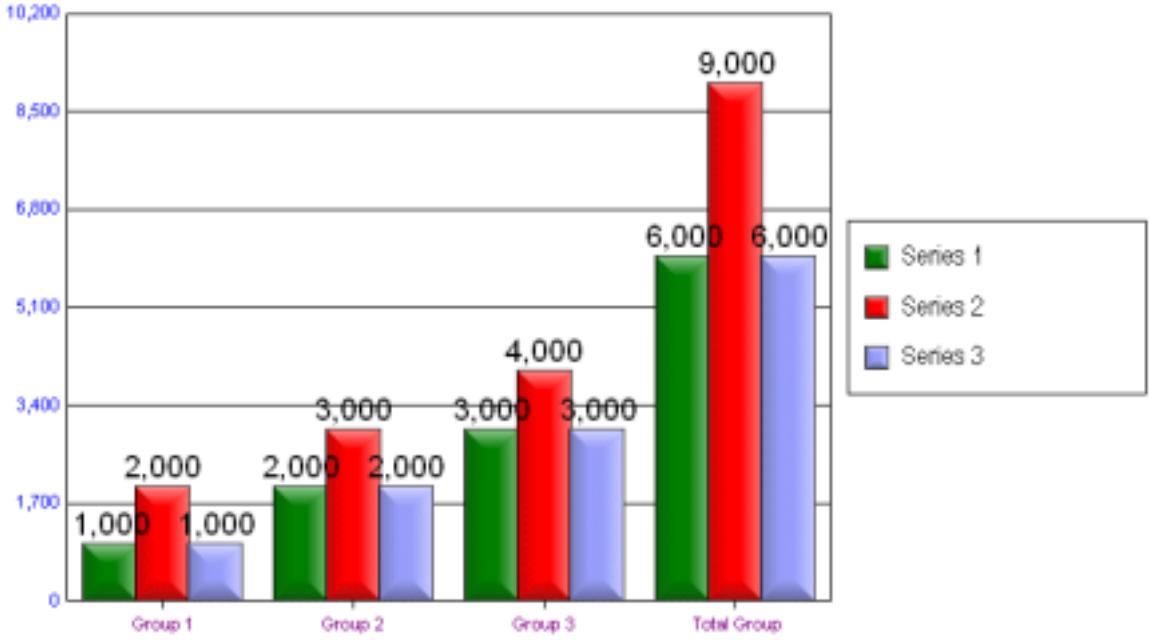
```
@TOTAL_GROUP szLabel
```

PARAMETERS:

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

EXAMPLE:

```
@TOTAL_GROUP Total Group
@BEVEL 3 1
@DATATEXT 1
```



PERSISTENT:

NO

@USER_SERIES (User-Defined Series)

This powerful 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 Crystal Reports and will therefore always be the last series in the legend.

SYNTAX:

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

PARAMETERS:

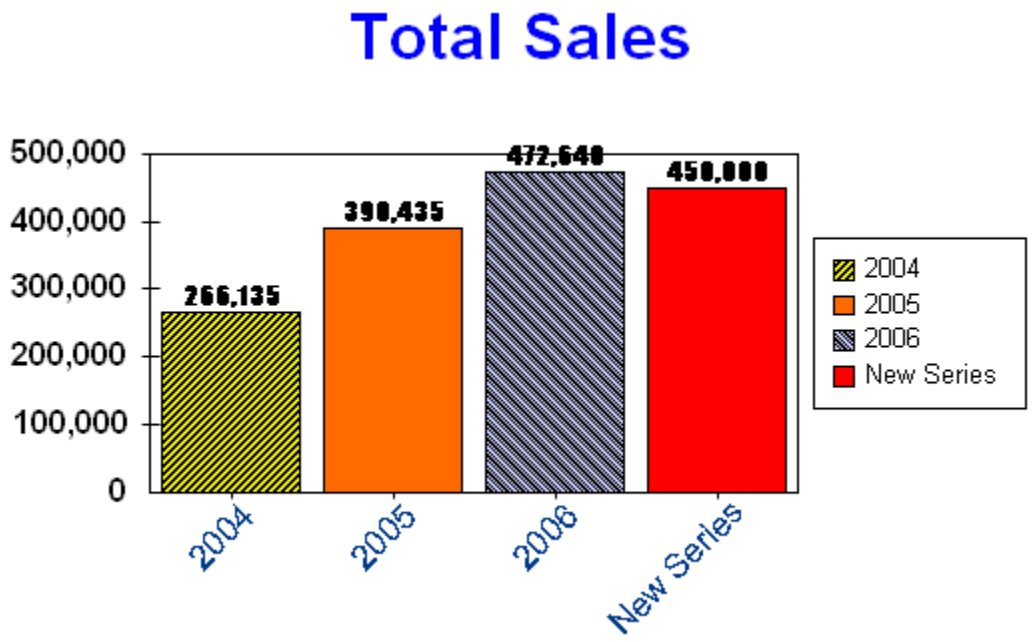
nElements; 1...1024 defines the number of *fValues* that follow. For example if *nElements* is 3, it must be followed by 3 *fValues* that will be assigned to the first 3 groups of the new series.

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 1 450000 New Series
```



PERSISTENT:

NO


THREE D GRAPHICS
CHARTS UNLIMITED

Section 5: Labels

Use these macros to manage series and group labels.

- @3DLABEL; Automatically arrange labels in a 3D Chart.
- @AGL; Alias/Change a Group Label
- @ASL; Alias/Change a Series Label
- @ASL_DP; Map a sub-string from a label to the legend
- @COND_GROUP_LABEL; Color a riser based on a Group Label
- @COND_GROUP_LABEL2; Color a riser based on a Group Label prefix
- @GROUP_LABELS_ON_BASELINE; Draw Group Labels on Baseline
- @XSKIP; Skip labels on the X-Axis
- @XSKIP2; Skip labels on the X-Axis and force last label
- @XSKIP3; Maximum X-axis labels/auto-adjust skip to match

@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:

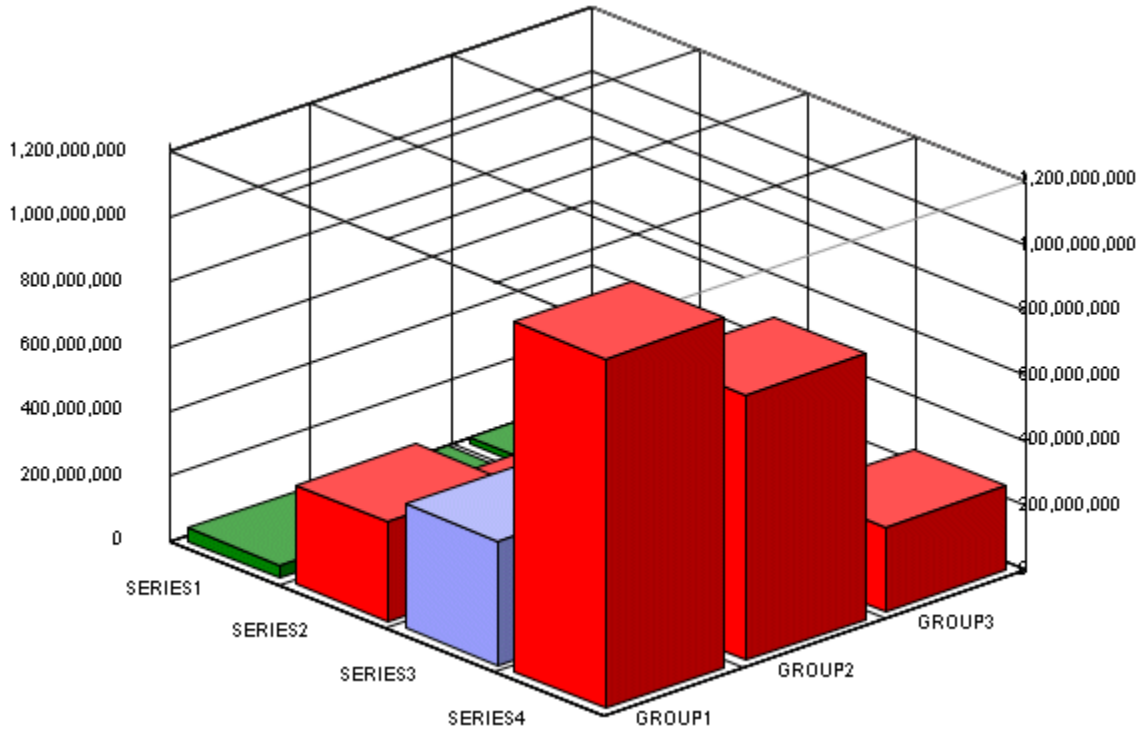
nLabel = 0...3 selects the labels to adjust
0 = Group Labels (normally along the right floor of the 3D cube)
1 = Series Labels (normally along the left floor of the 3D cube)
2 = Labels on the left side of the 3D cube
3 = Labels on the right side of the 3D cube

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 -500 0 @3DLABEL 1 -500 0
@3DLABEL 2 -500 0 @3DLABEL 3 -500 0
```



PERSISTENT:

YES

@AGL (Alias Group Label)

This macro can be used to change a group label.

SYNTAX:

```
@AGL nGroup sZLabel
```

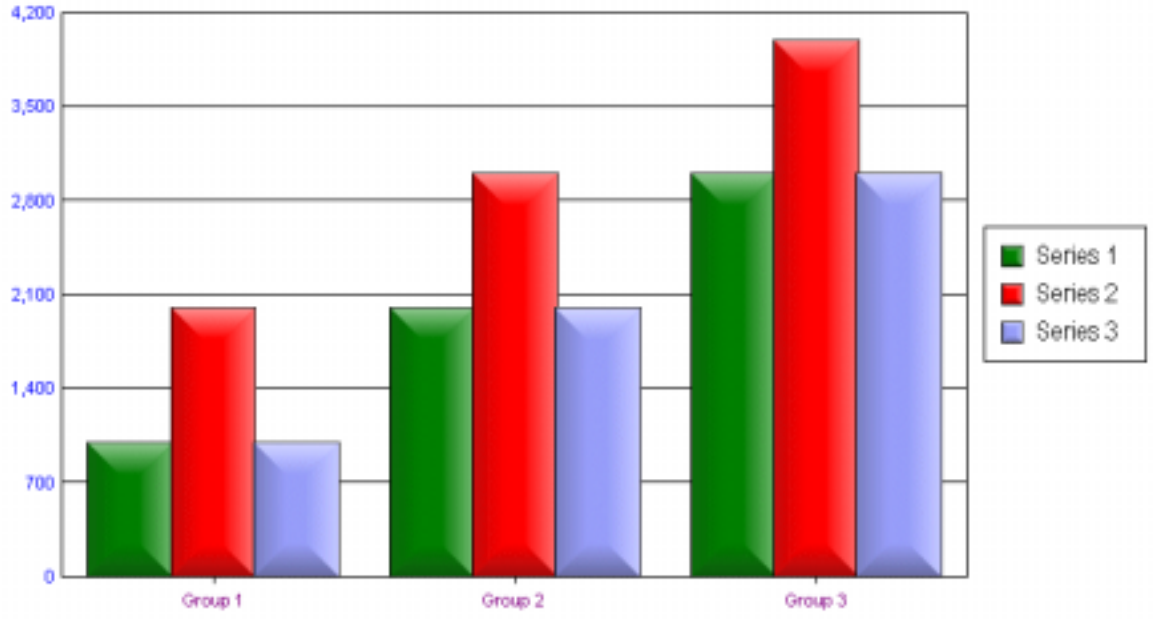
PARAMETERS:

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 1
@AGL 1 GROUP 2
@AGL 2 GROUP 3
@ASL 0 SERIES 1
@ASL 1 SERIES 2
@ASL 2 SERIES 3
```



PERSISTENT:

NO

ALSO SEE:

@ASL

@ASL (Alias Series Label)

This macro can be used to change a series label.

SYNTAX:

```
@ASL nSeries sZLabel
```

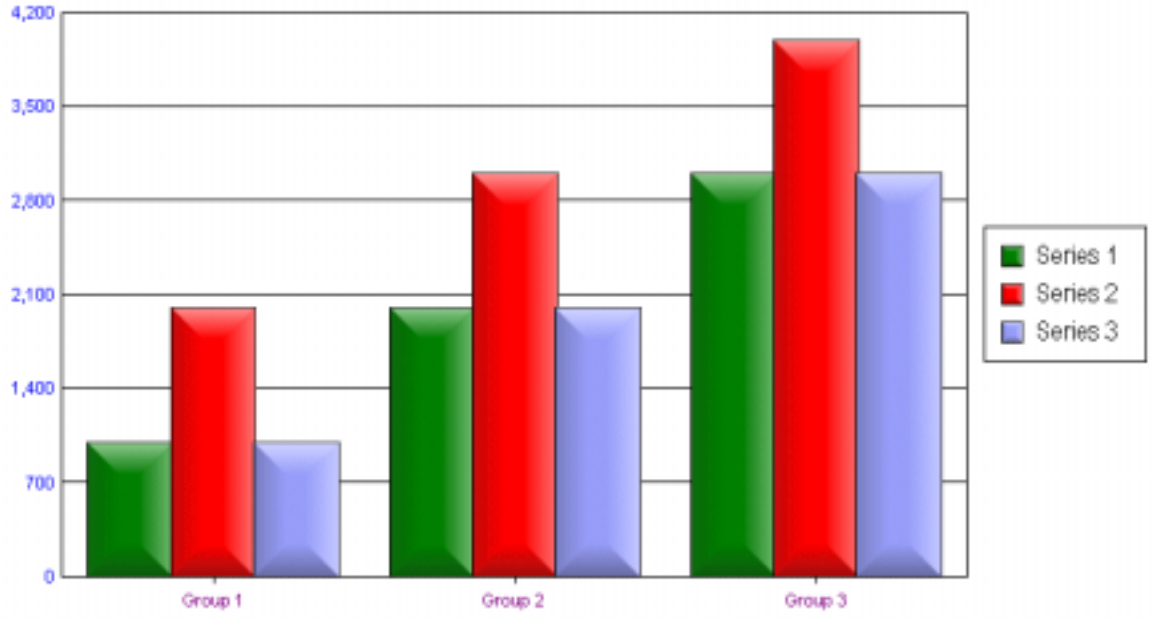
PARAMETERS:

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:

```
@AGL 0 GROUP 1
@AGL 1 GROUP 2
@AGL 2 GROUP 3
@ASL 0 SERIES 1
@ASL 1 SERIES 2
@ASL 2 SERIES 3
```



PERSISTENT:

NO

ALSO SEE:

@AGL

@ASL_DP (Alias Series Label/Data Point)

If the first group label contains a tilde (~), this macro maps the sub-string to the left of the first tilde to the series (legend) label specified by *nSeries*. When group labels are drawn, the sub-string and tilde are stripped out so that neither appear on the group axis.

SYNTAX:

```
@ASL_DP nSeries
```

PARAMETERS:

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

PERSISTENT:

NO

ALSO SEE:

@ASL

@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:

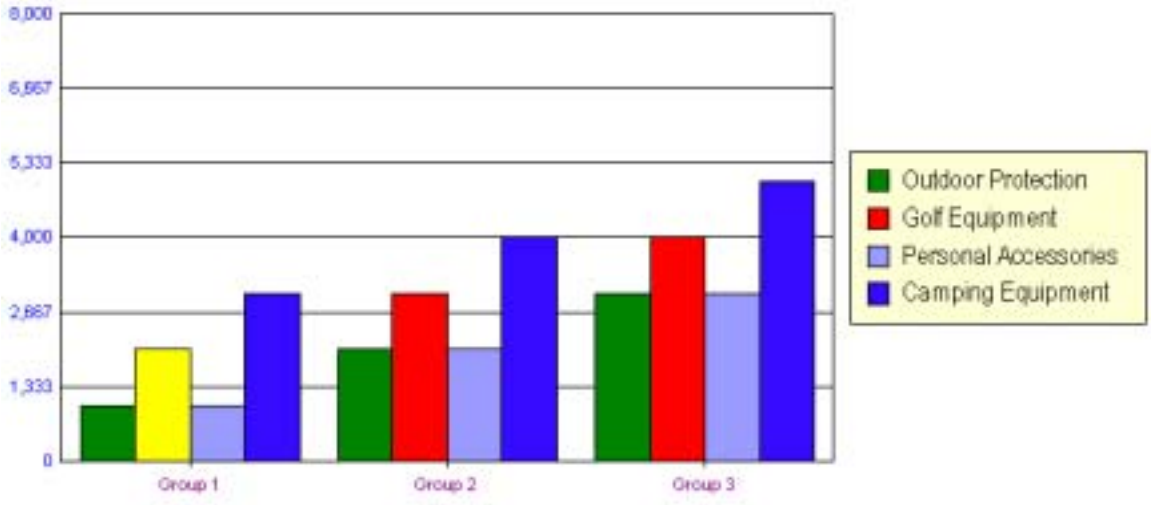
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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

szGroupLabel; Group label string

EXAMPLE:

```
@COND_GROUP_LABEL 1 255 255 0 Group 1
```



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 (*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.

PERSISTENT:

NO

@GROUP_LABELS_ON_BASELINE (Group Labels on Baseline)

This macro can be used to draw group labels on the baseline of the chart instead of the axis/frame.

SYNTAX:

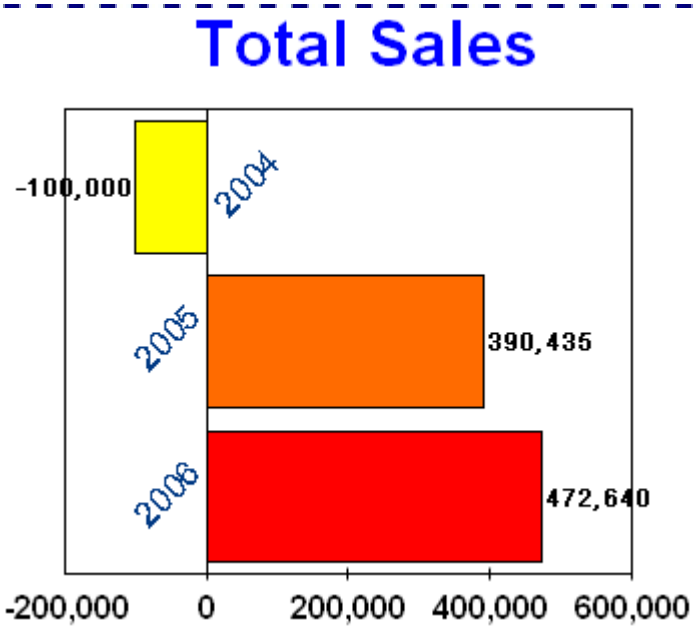
```
@GROUP_LABELS_ON_BASELINE bShowOnBaseLine
```

PARAMETERS:

bShowOnBaseLine; 0/1
0=Draw group labels normally on the axis/frame.
1=Draw group labels on the baseline of the chart.

SYNTAX:

```
@GROUP_LABELS_ON_BASELINE 1
```



PERSISTENT:

YES

@XSKIP (X-Axis Skip Labels)

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

SYNTAX:

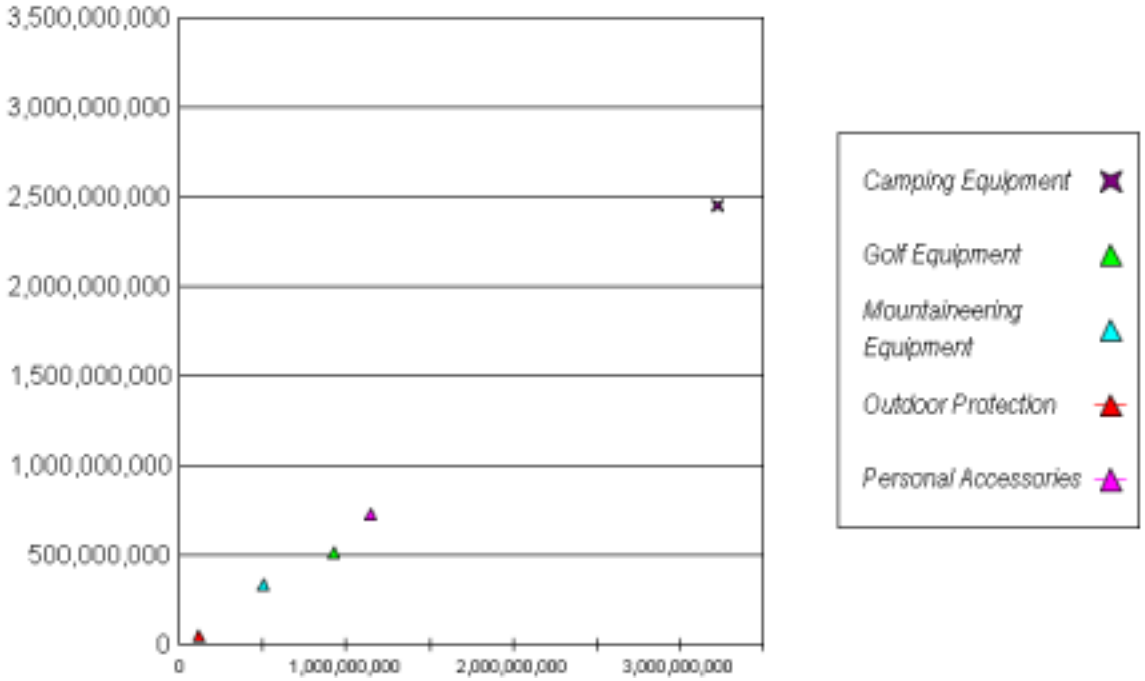
```
@XSKIP nSkip
```

PARAMETERS:

nSkip; skip value

EXAMPLE:

```
@XSKIP 1
```



PERSISTENT:

NO

ALSO SEE:

@XSKIP2, @XSKIP3

@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

PERSISTENT:

NO

ALSO SEE:

@XSKIP, @XSKIP3

@XSKIP3 (Max X-Axis Labels/Auto Adjust Skip)

This macro specifies a maximum number of X-axis labels and auto-adjusts the skip to match.

SYNTAX:

```
@XSKIP3 nNumLabels
```

PARAMETERS:

nNumLabels; specifies the maximum number of labels

PERSISTENT:

NO

ALSO SEE:

@XSKIP, @XSKIP2


THREE D GRAPHICS
CHARTS UNLIMITED

Section 6: Riser & Markers

Use these macros to control and format risers and markers:

- @BUBBLEMODE; Control appearance of bubbles in a bubble chart
- @HIDE_ZERO; Hide Risers/Markers for zero data points
- @HL; Highlight a Riser/Marker
- @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

@BUBBLEMODE (Bubble Size in Bubble Charts)

This macro controls the appearance of bubbles in a bubble chart. When bubble mode is activated (*bActivate*=1), bubble sizes are calculated using relative area. If the largest bubble represents a value of 100 and the smallest represents a value of 1 (for example), the smallest bubble will be 1/100th the area of the largest bubble. A bubble that represents a negative value will be drawn using the absolute value. In a data set such as 1, 50, -100, the -100 value is drawn as the largest bubble. When bubble mode is disabled (*bActivate*=0), bubble sizes are calculated using a linear interpolation between the maximum and minimum. Negative values are drawn as smaller bubbles. In a data set such as 1, 50, -100, the -100 value is drawn as the smallest bubble.

SYNTAX:

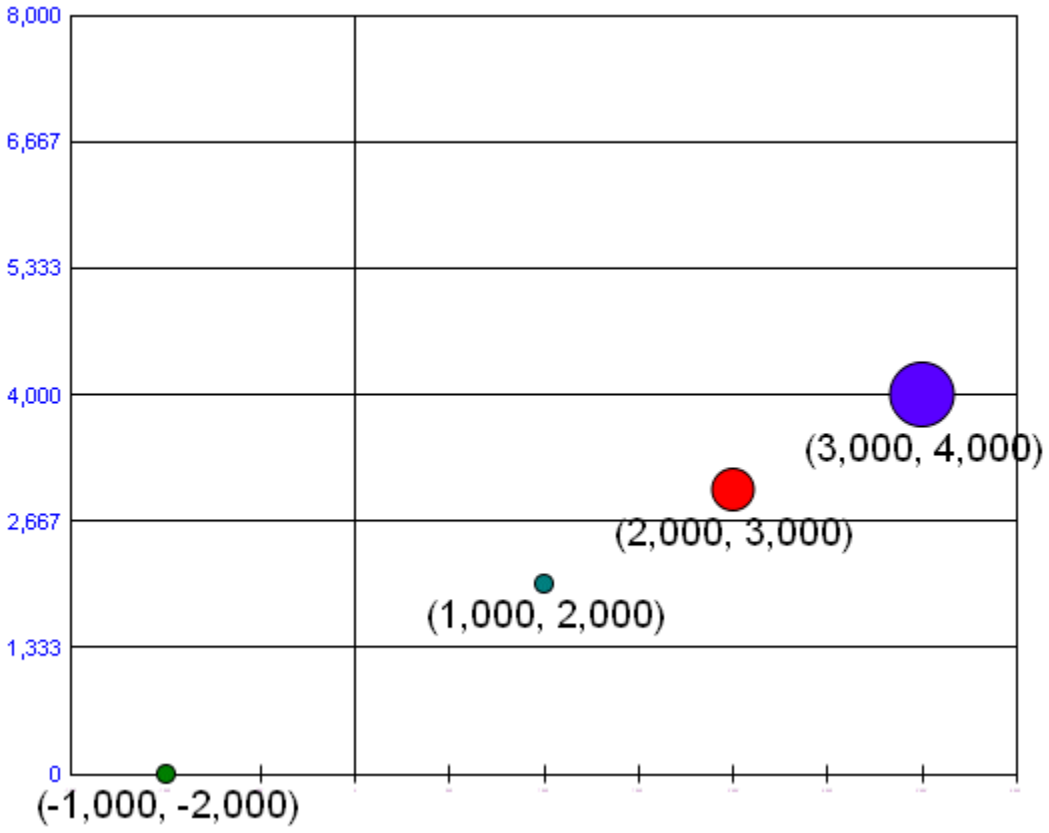
```
@BUBBLEMODE bActivate
```

PARAMETERS:

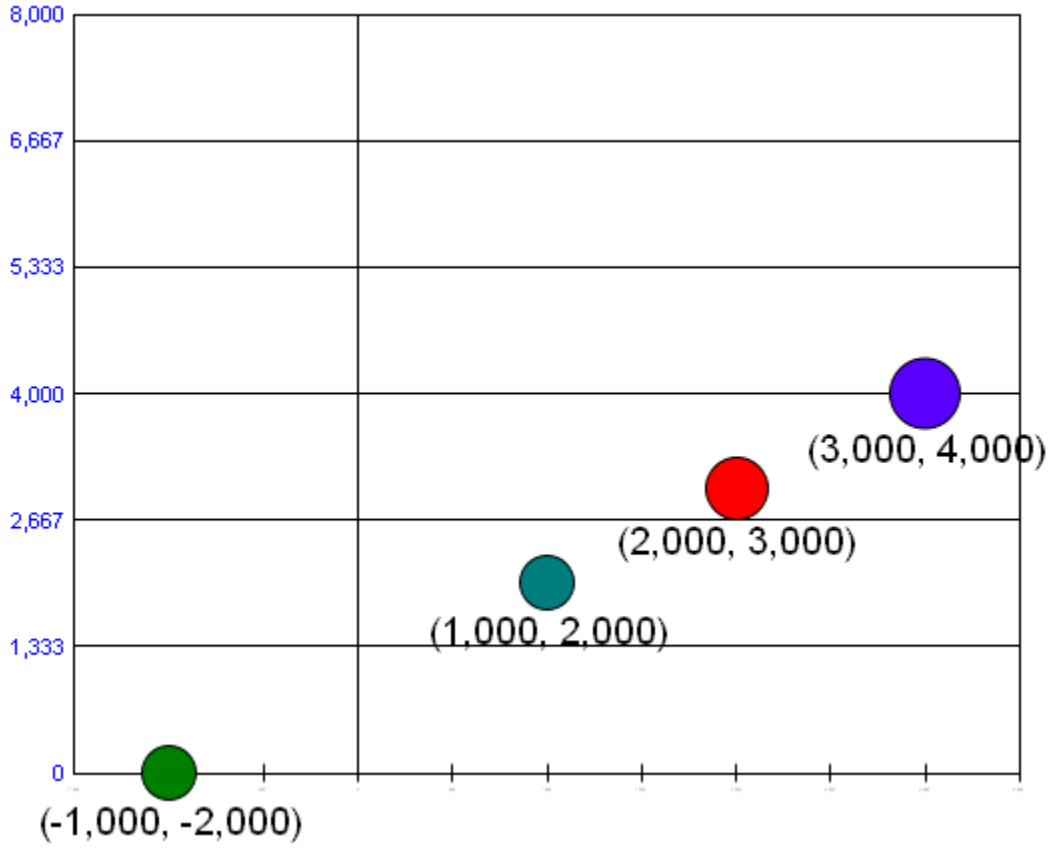
bActivate; 0=use normal Crystal Reports bubble size calculations, 1=activate Excel-Style bubbles

EXAMPLE:

```
@BUBBLEMODE 0
```



```
@BUBBLEMODE 1
```



PERSISTENT:

NO

@HIDE_ZERO (Hide Zero Riser/Marker)

This macro hides any riser or marker in a bar, line, area, 3D, or pie chart that is equal to 0.0. The value becomes “NULL Data” for purposes of all chart calculations and output.

SYNTAX:

```
@HIDE_ZERO
```

PARAMETERS:

None

PERSISTENT:

NO

@HL (Highlight Riser/Marker)

This macro uniquely colors the riser or marker identified by *nSeries* and *nGroup* with the color specified by *nRed*, *nGreen*, *nBlue*.

SYNTAX:

```
@HL nSeries nGroup nRed nGreen nBlue
```

PARAMETERS:

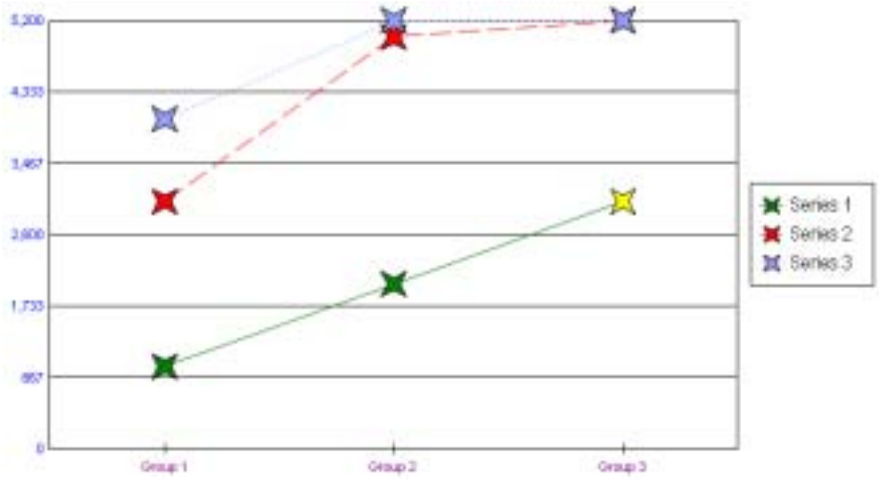
nSeries; Series # (0=Series 1)

nGroup; Group # (0=Group 1)

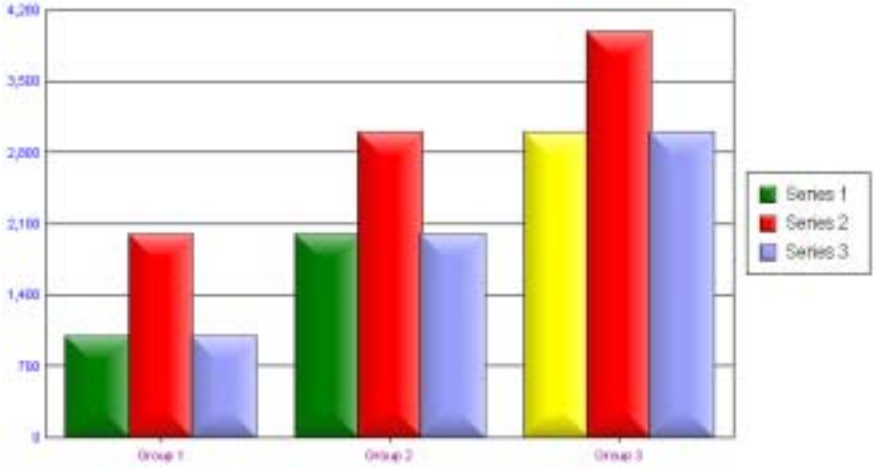
nRed, nGreen, nBlue: 0...255 specifies the RGB value to be applied to the riser/marker at *nSeries* and *nGroup*.

EXAMPLE:

```
@HL 0 2 255 255 0
```




```
@GRAPHTYPE 14
@HL 0 2 255 255 0
```



PERSISTENT:

YES

@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.

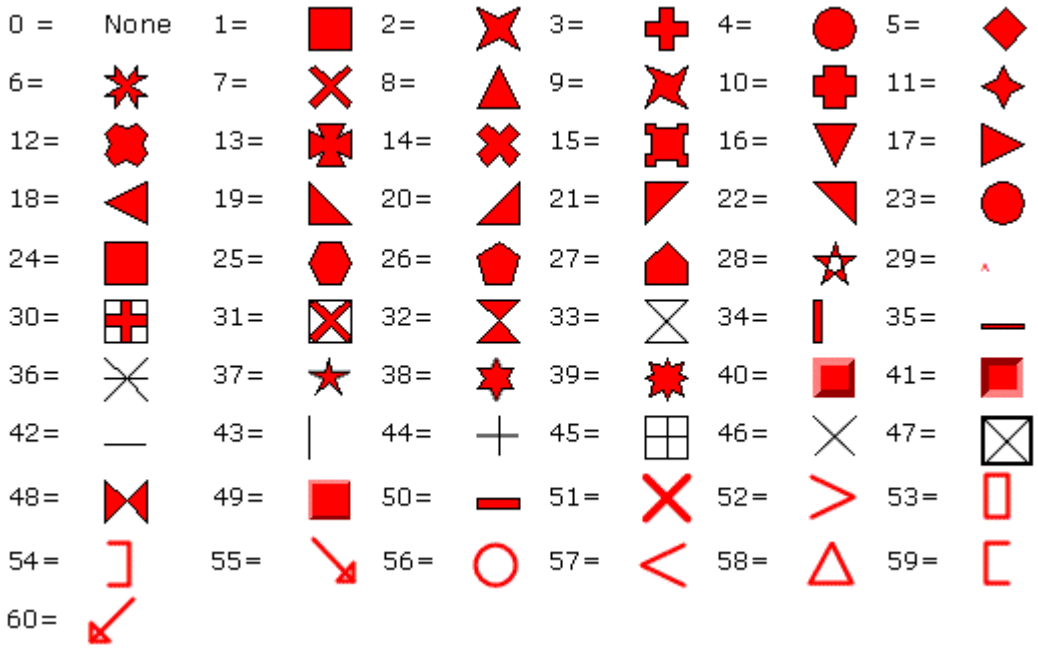
SYNTAX:

```
@MARKER nSeries nMarker
```

PARAMETERS:

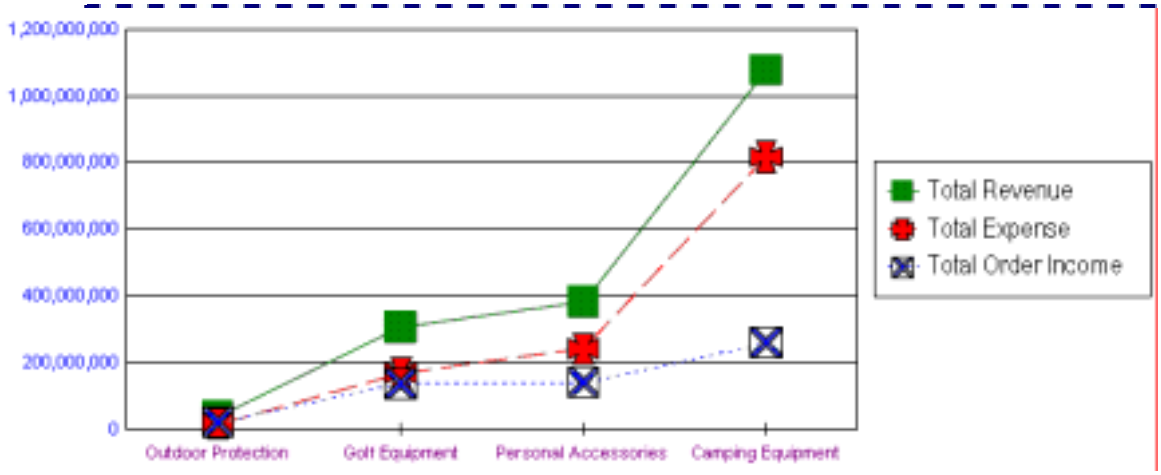
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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



EXAMPLE:

```
@SZ 50 @MARKER 0 40 @MARKER 1 13 @MARKER 2 31
```



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.

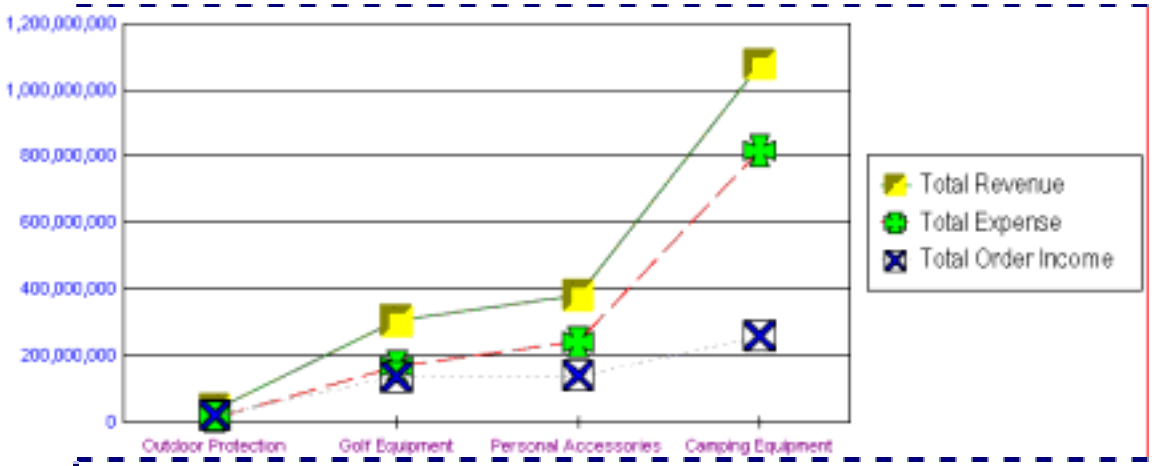
nRed; 0...255 defines the Red portion of RGB color selection.

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

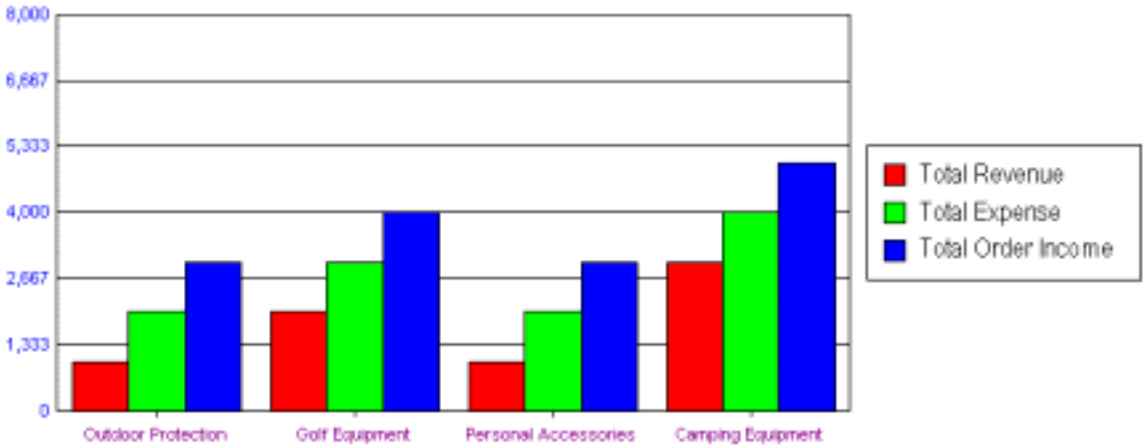
nBlue; 0...255 defines the Blue portion of RGB color selection.

EXAMPLE:

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




```
@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.

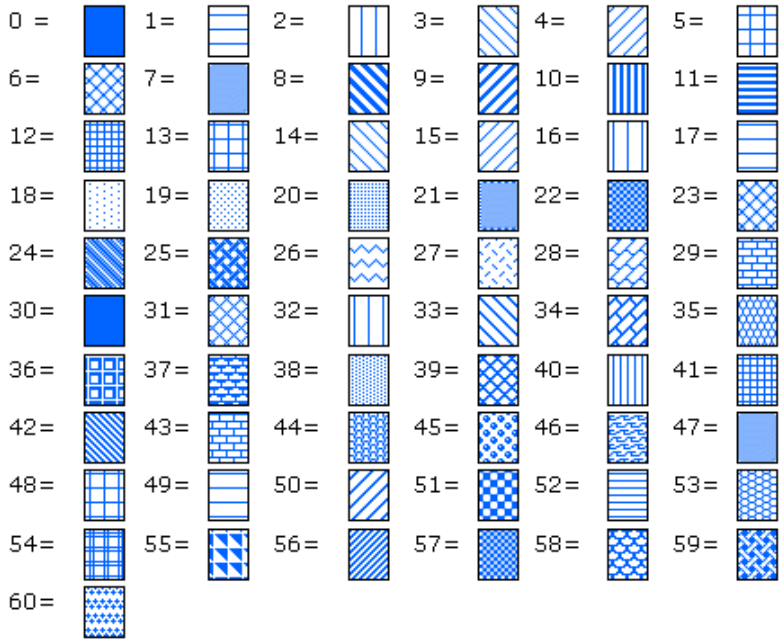
SYNTAX:

```
@PAT nSeries nPattern
```

PARAMETERS:

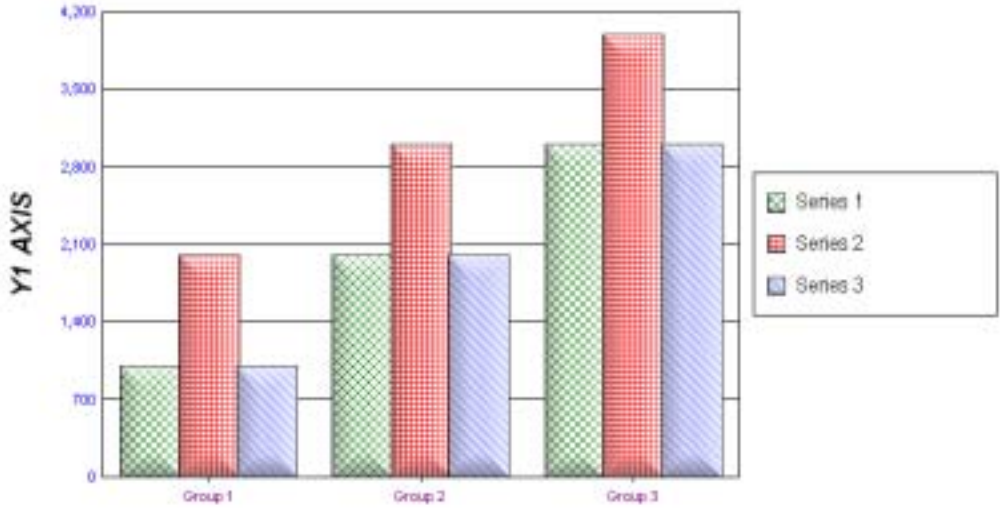
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nPattern; -60...60. Selects one of the following patterns with a white background. Negative values select a transparent background.



EXAMPLE:

```
@PAT 0 6 @PAT 1 12 @PAT 2 24
```



PERSISTENT:

YES

@RISER_BORDER (Riser Border)

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

SYNTAX:

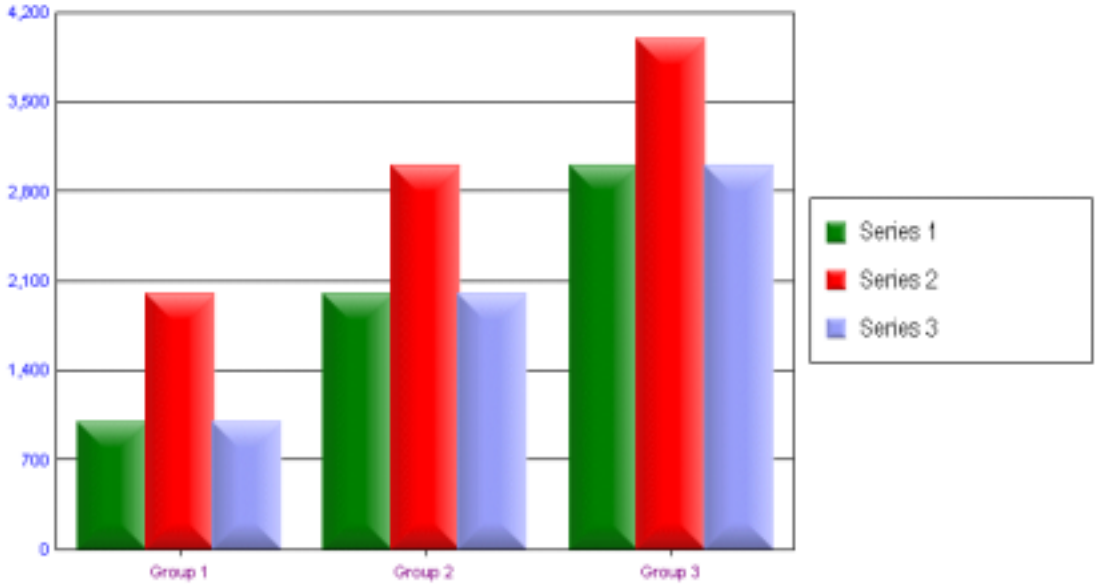
```
@RISER_BORDER bShow
```

PARAMETERS:

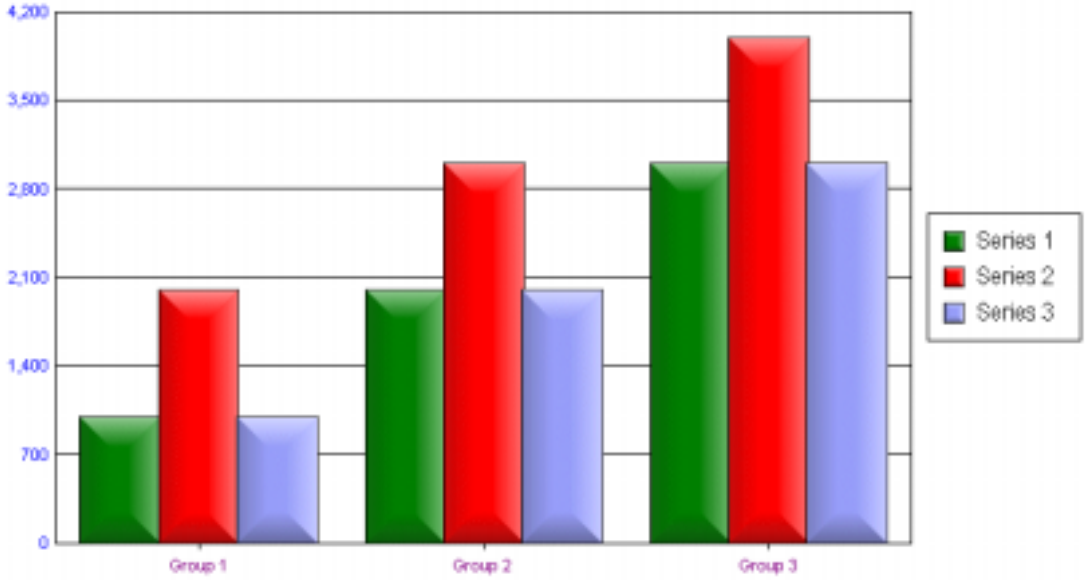
bShow; 0 = Turn OFF border line on risers/markers, 1= Turn ON border line on risers/markers.

EXAMPLE:

```
@RISER_BORDER 0
```




```
@RISER_BORDER 1
```



PERSISTENT:

YES

@RISER_OVERLAP (Riser Overlap)

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

SYNTAX:

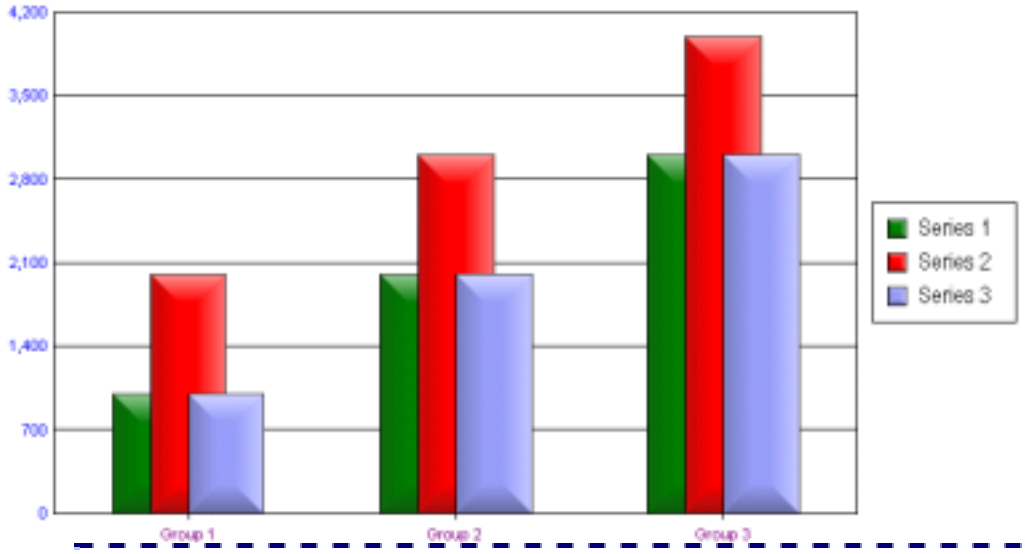
```
@RISER_OVERLAP nOverlap
```

PARAMETERS:

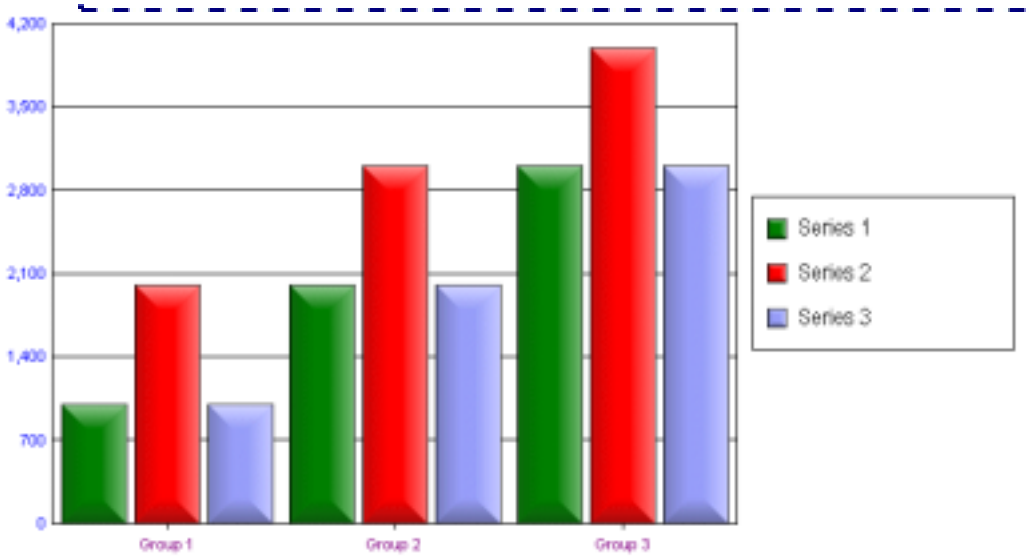
nOverlap; -100...100 selects the amount of overlap. Negative numbers make the risers overlap. A value of -100 will place all risers in a group draw on top of each other. Larger numbers will draw the risers farther apart.

EXAMPLE:

```
@RISER_OVERLAP -50
```




```
@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:

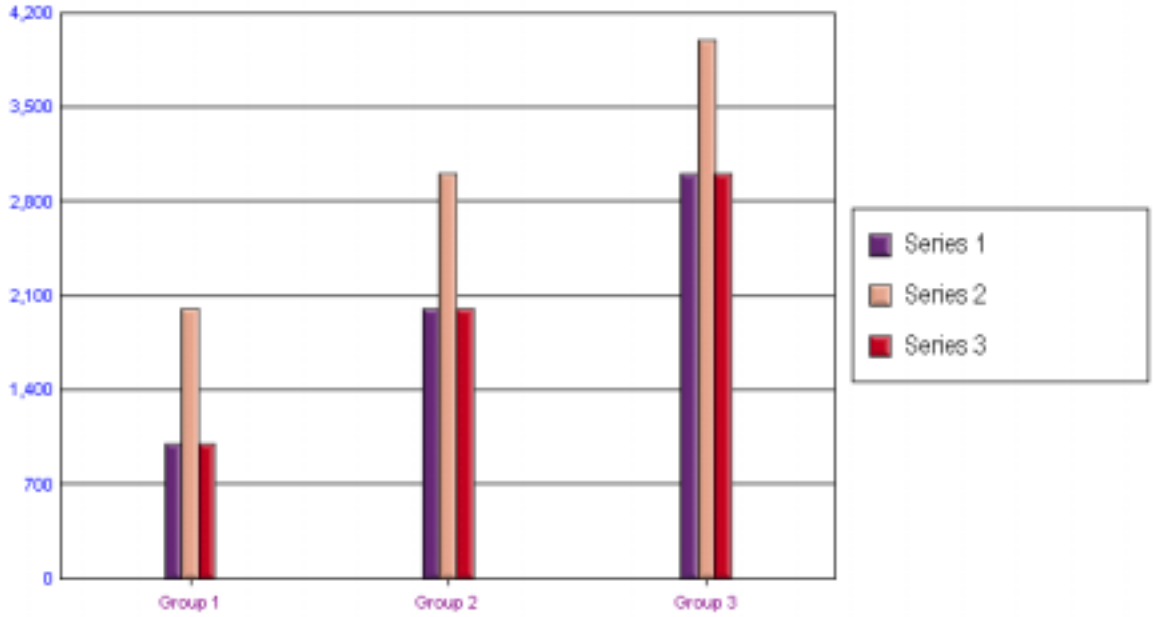
```
@RISER_WIDTH nWidth
```

PARAMETERS:

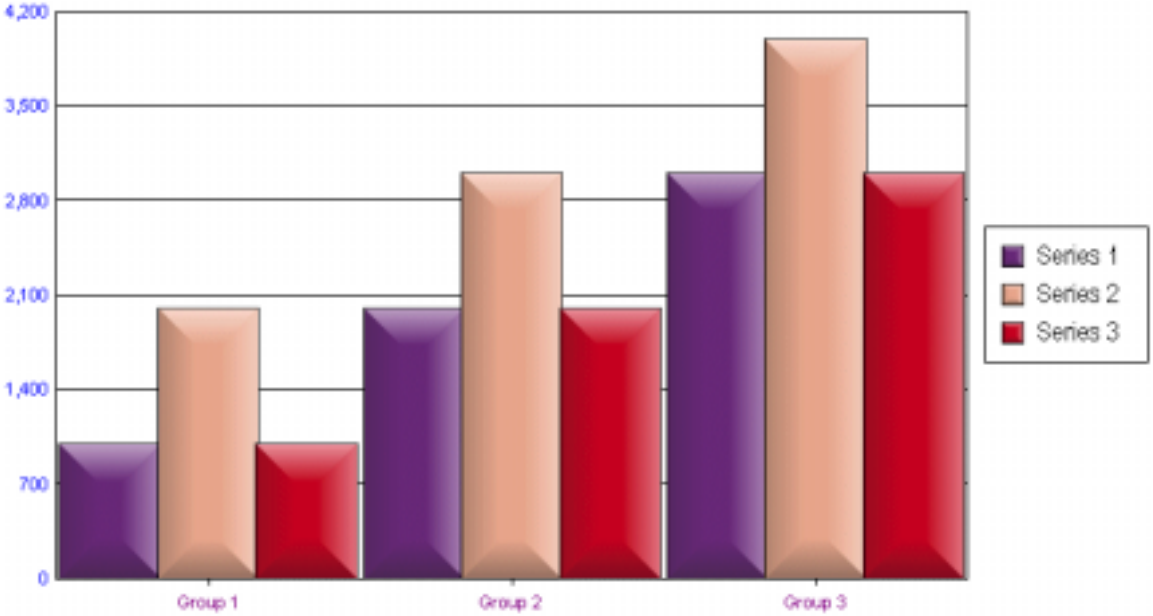
nWidth; 0...100 selects the width of risers.

EXAMPLE:

```
@RISER_WIDTH 20
```




```
@RISER_WIDTH 100
```



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:

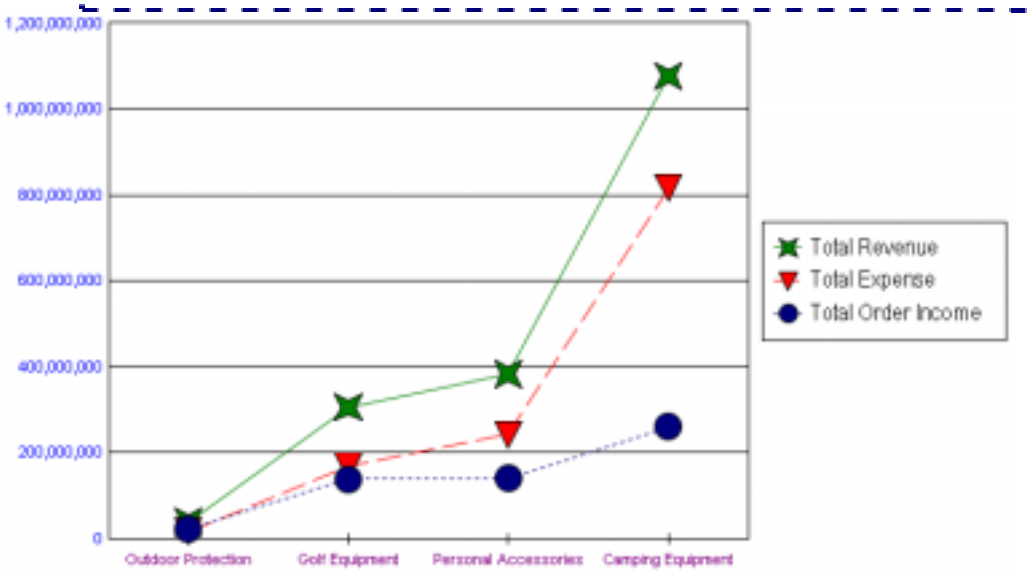
```
@SZ nValue
```

PARAMETERS:

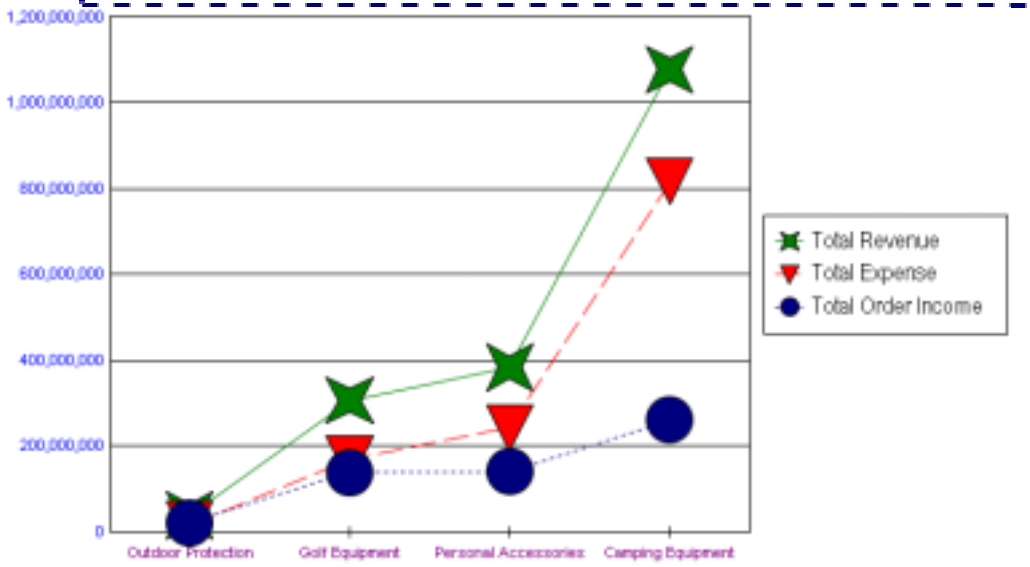
nValue; Size of markers (1...100)

EXAMPLE:

```
@SZ 50
```




```
@SZ 100
```



PERSISTENT:

NO

ALSO SEE:

@MARKER, @MS & @MC

Section 7: Data & Data Text

Use these macros to manage data and data text:

- @APPEND_DATATEXT; Append a String to Data Text
- @DATATEXT; Select a Data Text Mode
- @DLT; Data Line Type
- @DP; Data Point Override
- @DPC; Data Point Clear
- @DT; Show/Hide Series-Specific Data Text
- @FORCE_DATALINE; Force data line connecting all data points
- @FORCE_DATATEXT_CURRENCY; Force data text to currency format
- @IR; Insert Row (i.e., create a user-defined series of data)
- @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

@APPEND_DATATEXT (Append String to Data Text)

This macro can be used to append a string to each piece of data text in a chart.

SYNTAX:

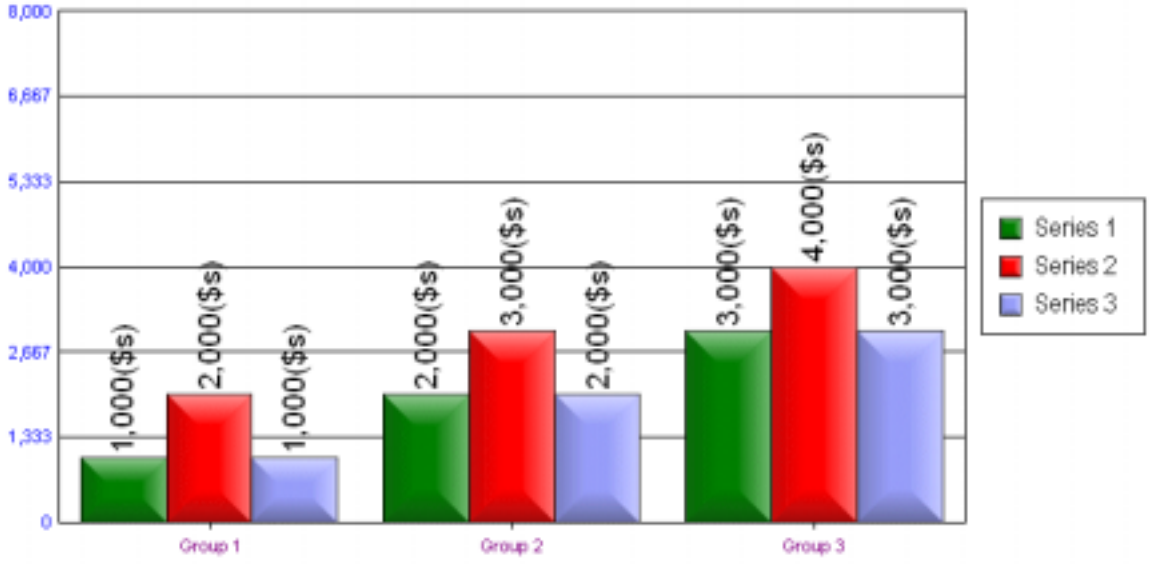
```
@APPEND_DATATEXT szString
```

PARAMETERS:

szString; String to append to data text. If you intend to define another macro in the same title field, terminate the label string with a '~'.

EXAMPLE:

```
@DATATEXT 1
@RDT 2
@APPEND_DATATEXT ($s)
```



PERSISTENT:

NO

@DATATEXT (Data Text Mode)

This macro sets the data text mode (i.e., values on risers, labels on risers, etc.).

SYNTAX:

@DATATEXT ***n***

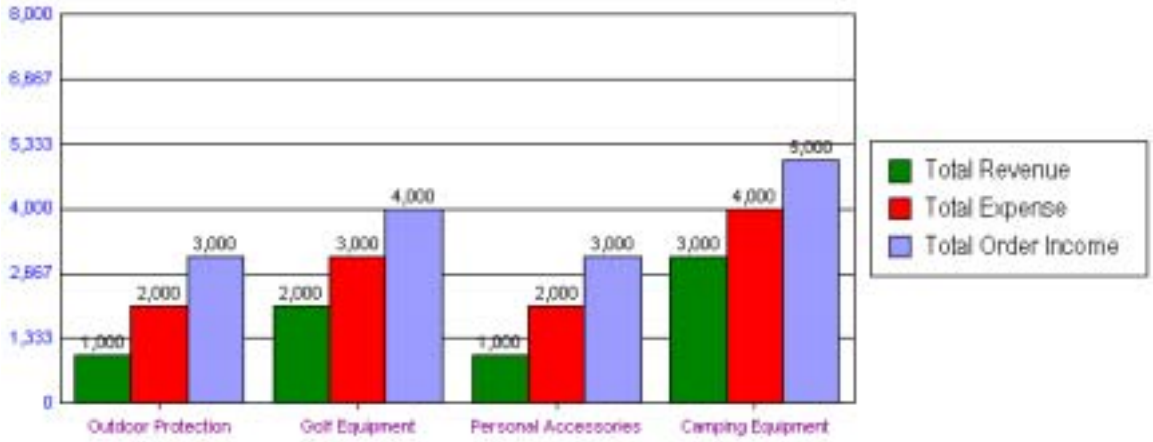
PARAMETERS:

n; Selects the data text to show. It can be one of the following:

Value	Description
0	None, do not show values or non-numeric text
1	Value(s) of data item(s).
2	Unique data item text.
3	Values and non-numeric text
4	Absolute value in stacked segments.
5	Unique data item text and show absolute value in stacked segments
7	Values and non-numeric text and show absolute value in stacked segments
8	Total value on top of the stacked bar.
16	Z-value in bubble charts.
32	Calculate and show the riser value as a percentage
64	Y-value in scatter or bubble charts

EXAMPLE:

```
@DATATEXT 1
```



PERSISTENT:

YES

@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
```

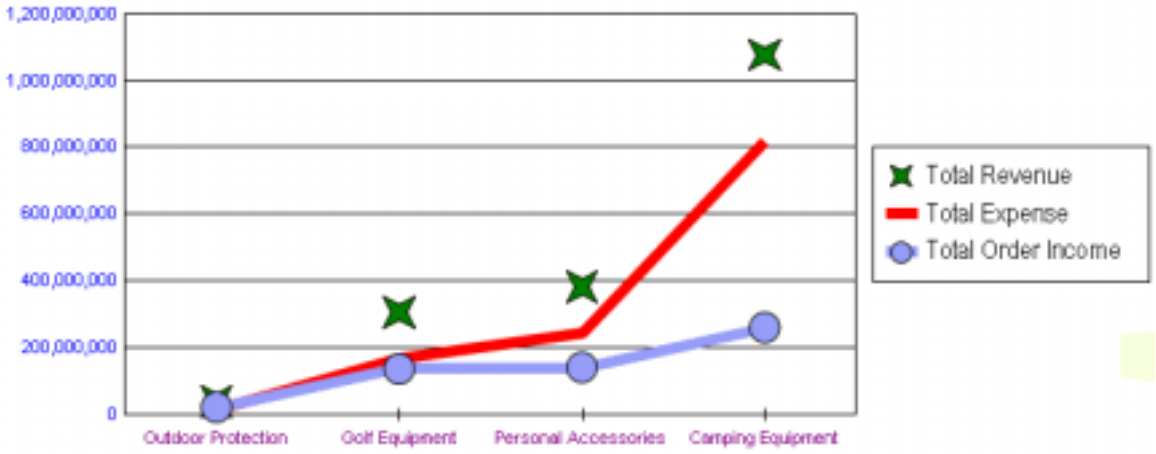
PARAMETERS:

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...3
1 = draw markers only
2 = draw lines only
3 = draw markers and lines.

EXAMPLE:

```
@LS -1 500 0
@SZ 50
@DLT 0 1
@DLT 1 2
@DLT 2 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:

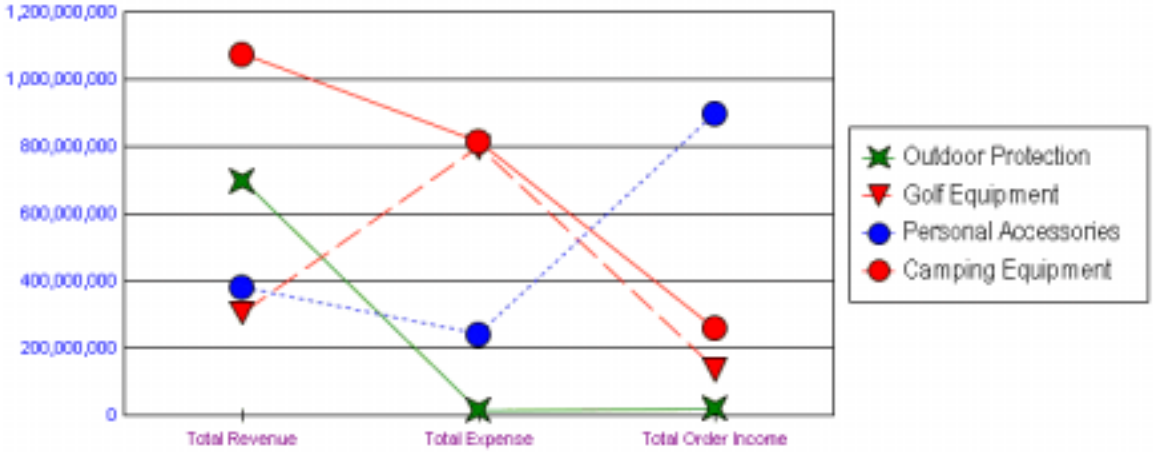
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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 700000000
@DP 1 1 800000000
@DP 2 2 900000000
```



Also See @FORECAST

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, are, or pie chart.

SYNTAX:

```
@DPC nSeries nGroup
```

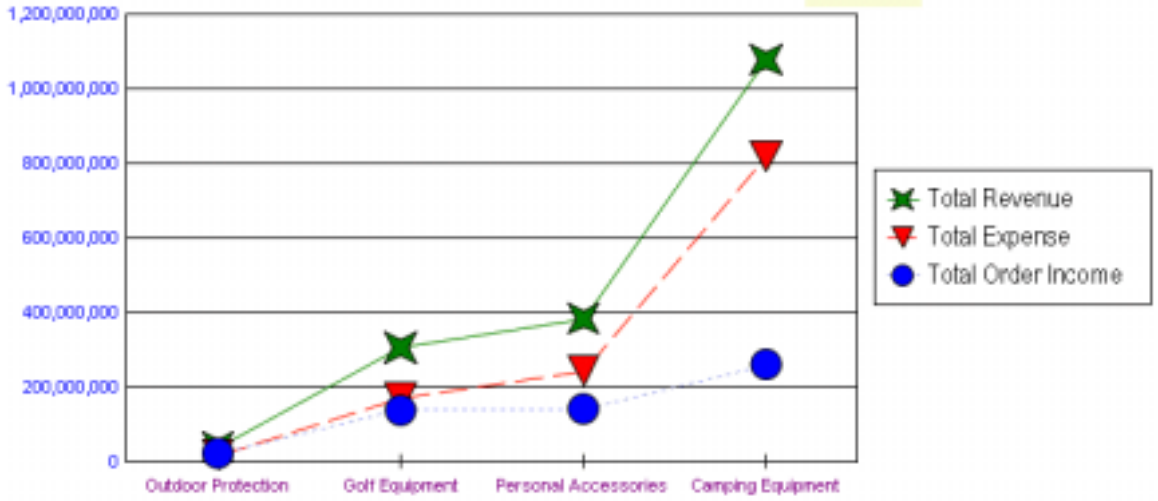
PARAMETERS:

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.

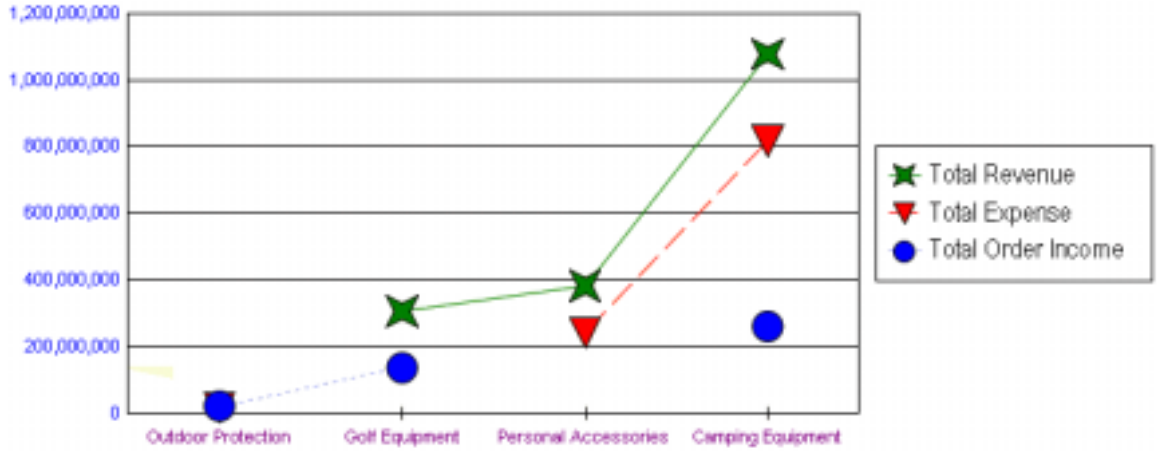
nGroup; -1...*n* (where: *n* = the total number of groups in the chart). -1 = apply to all groups, 0 = Group 1, 1 = Group 2, etc.

EXAMPLE:

```
CHART BEFRE @DPC
```




```
@DPC 0 0
@DPC 1 1
@DPC 2 2
```



PERSISTENT:

NO

ALSO SEE:

@DP

@DT (Show/Hide Series-Specific Data Text)

This macro can be used to show/hide data text for a specified series.

SYNTAX:

```
@DT nSeries bShow
```

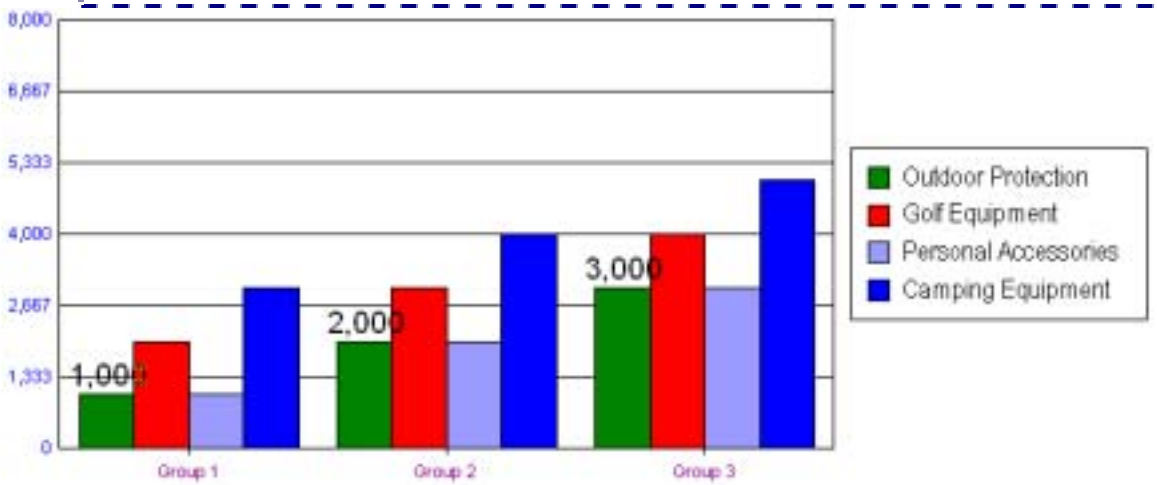
PARAMETERS:

nSeries; 0...511 specifies the series to show/hide data text (0=Series 1)

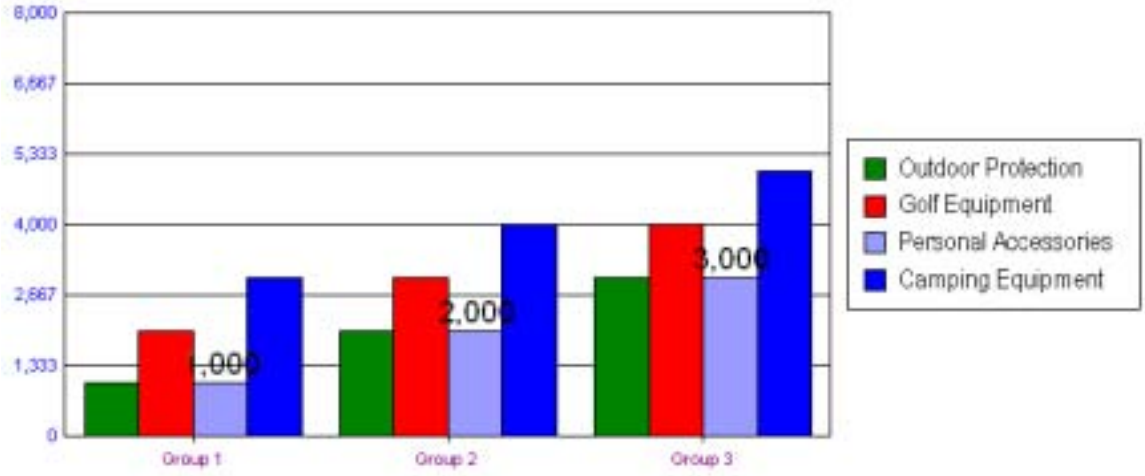
bShow; 1=Show Data Text, 0=Hide Data Text

EXAMPLE:

```
@DT 0 1
```




```
@DT 2 1
```



PERSISTENT:

YES

@FORCE_DATALINE (Force Data Line)

For scatter charts, this macro draws a data line connecting all data points even if the dataset implies one group and many series (i.e., each marker has a unique color).

SYNTAX:

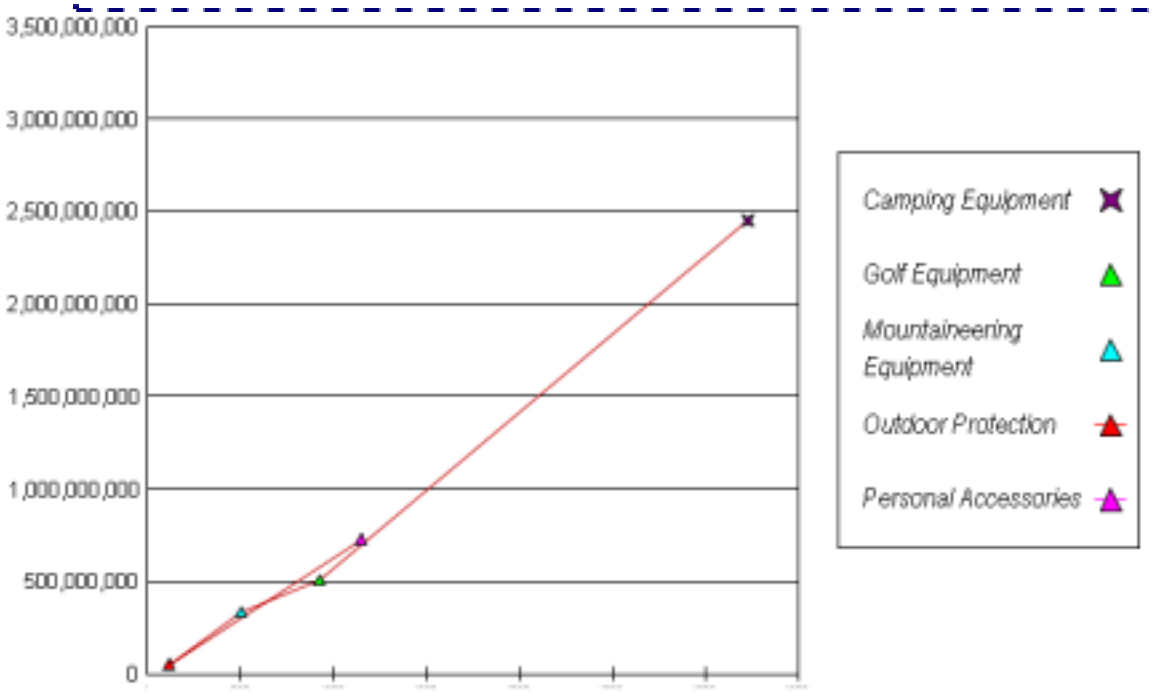
```
@FORCE_DATALINE nMode
```

PARAMETERS:

nMode; 0=Do nothing. 1=Connect all data points in a scatter chart. 2=Connect all data points in a scatter chart, but use sequential legend entries with identical strings to determine where to put line breaks.

EXAMPLE:

```
@FORCE_DATALINE 1
```



PERSISTENT:

NO

@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 0
```

PERSISTENT:

YES

@IR (Insert Row)

This macro creates a user-defined series of data.

SYNTAX:

```
@IR nSeries nElements [fValue1 fValue2 ... fValueN] szSeriesName~
```

PARAMETERS:

nSeries; Series to insert row. –1 = append to end of data set.

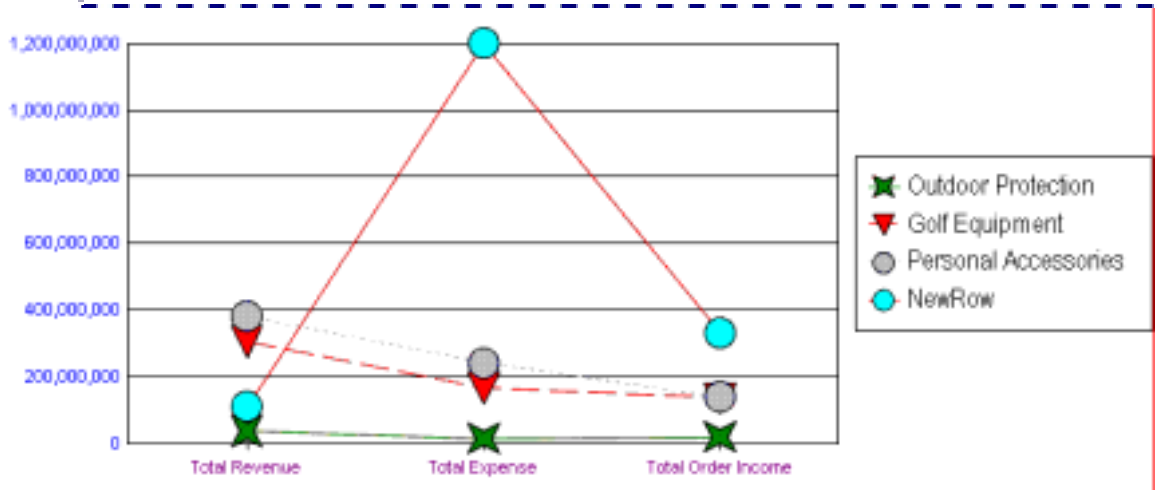
nElements; 1...1024 defines the number of *fValues* that follow. For example if *nElements* is 3, it must be followed by 3 *fValues* that will be assigned to the first 3 groups of the new series.

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

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

EXAMPLE:

```
@IR -1 3 111111111.1 2222222222.2 333333333.3 NewRow
@MCOLOR 3 87 251 226
```



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:

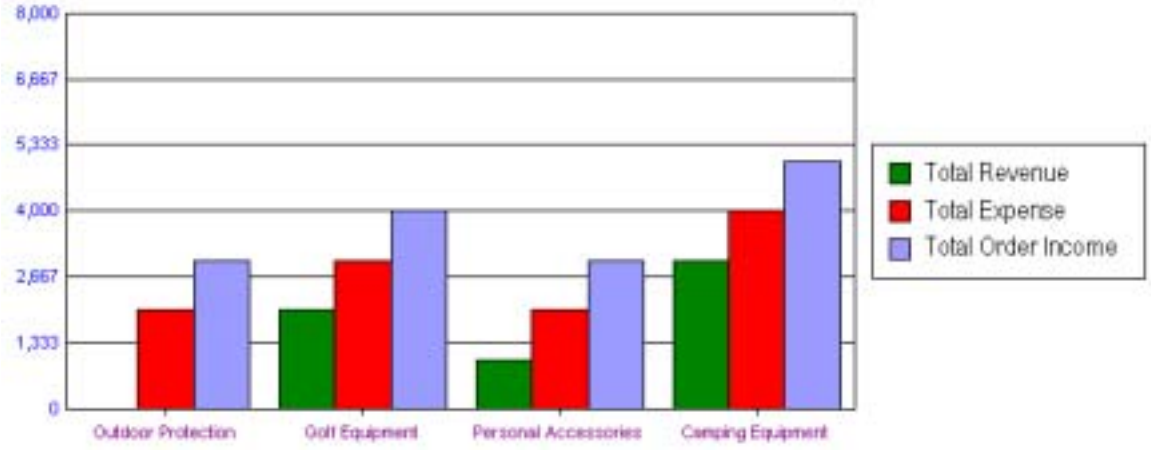
```
@OFFSCALE_Y1 nOffScaleMode
```

PARAMETERS:

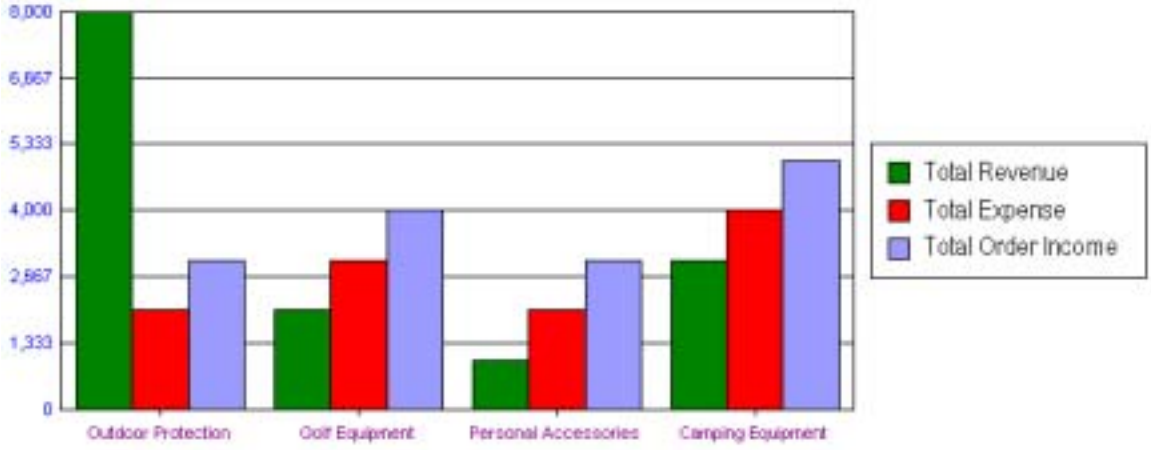
nOffScaleMode; 0/1. 0=Do not show off-scale values, 1=Graph off-scale values at scale maximum.

EXAMPLE:

```
@OFFSCALE_Y1 0
```




```
@OFFSCALE_Y1 1
```



PERSISTENT:

NO

@RDT (Rotate Data Text)

This macro can be used to rotate data text.

SYNTAX:

```
@RDT nRotateMode
```

PARAMTERS:

nRotateMode; 0, 2...6 selects one of the following:

0 = No rotation

2 = Rotate data text 90 degrees

3 = Rotate data text 180 degrees

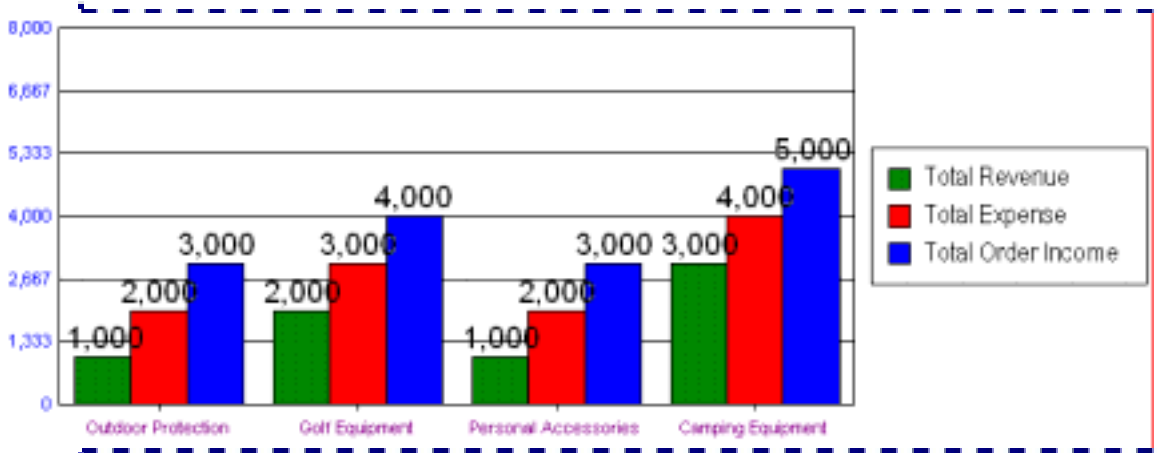
4 = Rotate data text 270 degrees

5 = Rotate data text 45 degrees

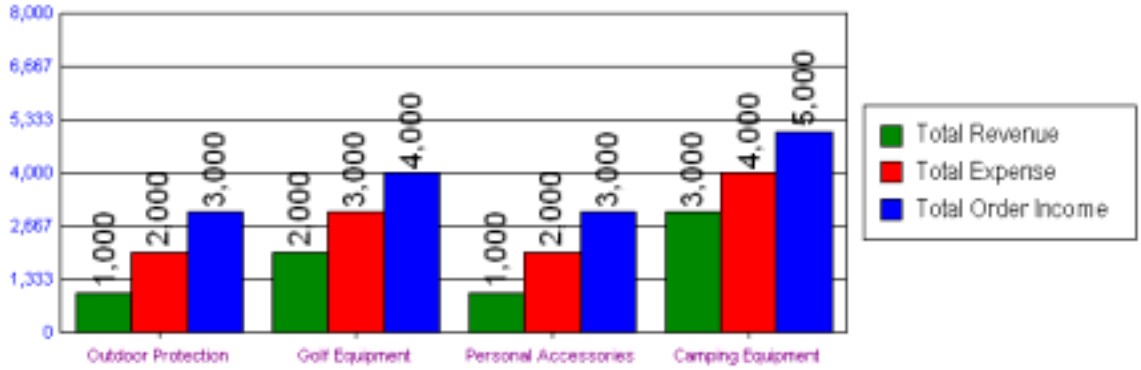
6 = Rotate data text 315 degrees

EXAMPLE:

```
@RDT 0
```




```
@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:

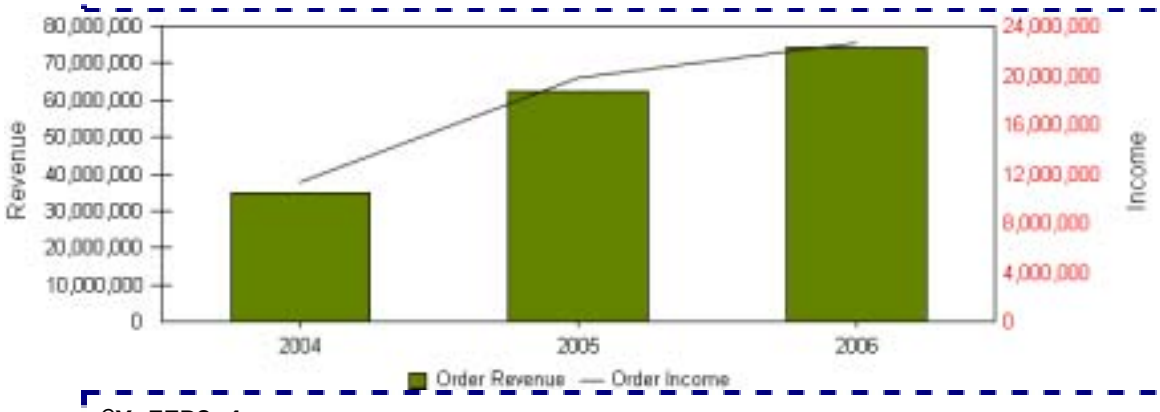
```
@Y_ZERO bExcludeZero
```

PARAMETERS:

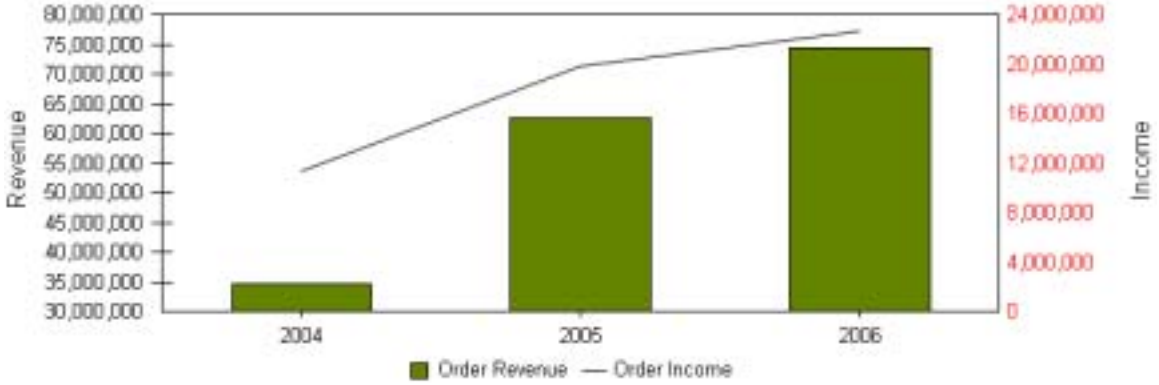
bExcludeZero; 1 = Zero is NOT used in calculating Y1-axis auto-scale. 0=Zero IS used in calculating Y1-axis auto-scale

EXAMPLE:

```
@Y_ZERO 0
```




```
@Y_ZERO 1
```



PERSISTENT:

NO


THREE D GRAPHICS
CHARTS UNLIMITED

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
- @ALT_FRAME_FILL; Alternating Frame Fill
- @ALT_FRAME_FILL_MODE; Alternating Frame Fill Mode
- @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
- @SHADOW; Apply a Drop Shadow to a chart object

Also see the @MC macro in "Box Plot Macros" (Section 12) to define markers colors in Box Plots.

Also see the @MCOLOR and @PAT macros in "Risers & Markers" (Section 6) 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
```

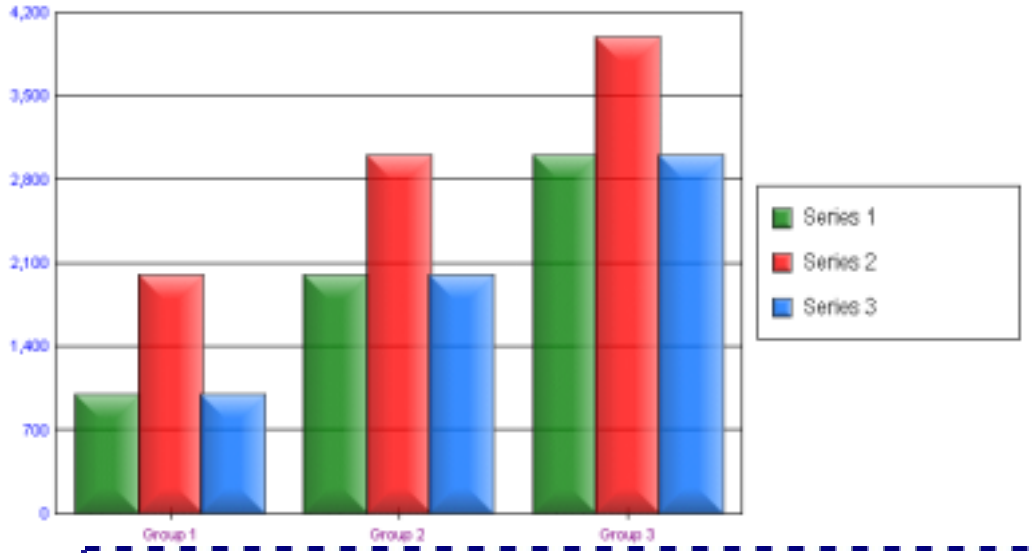
PARAMETERS:

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.

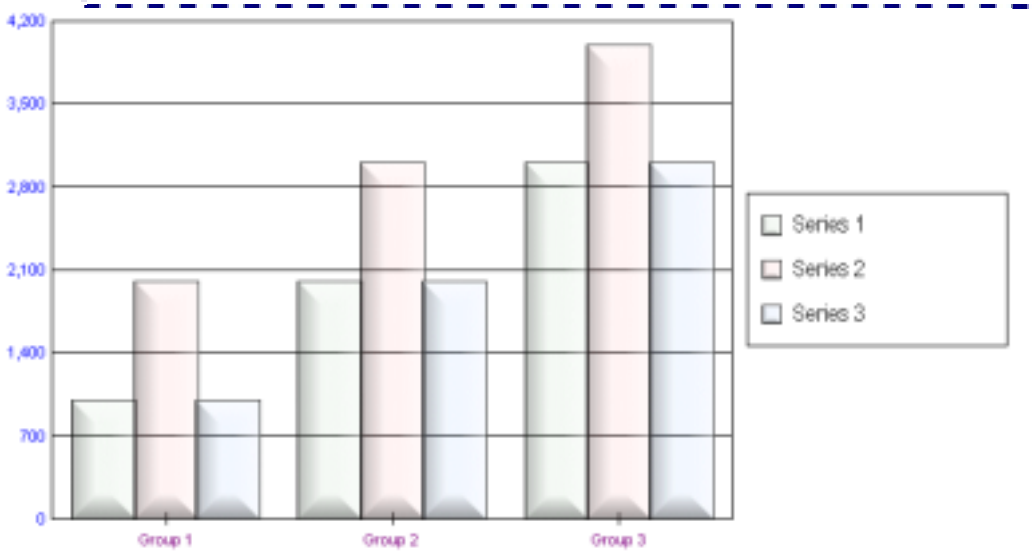
nValue; 0...255 selects the transparency level. 255 (the default) = no transparency. 0 = fully transparent.

EXAMPLE:

```
@ALPHA -1 200
```




```
@ALPHA -1 10
```



PERSISTENT:

YES

@ALT_FRAME_FILL (Alternating Frame Fill)

When @ALT_FRAME_FILL_MODE is 1 or 2, this macro colors the chart frame with alternating color bands based on the grid line divisions.

SYNTAX:

```
@ALT_FRAME_FILL nRed nGreen nBlue nAlpha
```

PARAMETERS:

nRed: 0...255 defines the Red portion of RGB fill color.

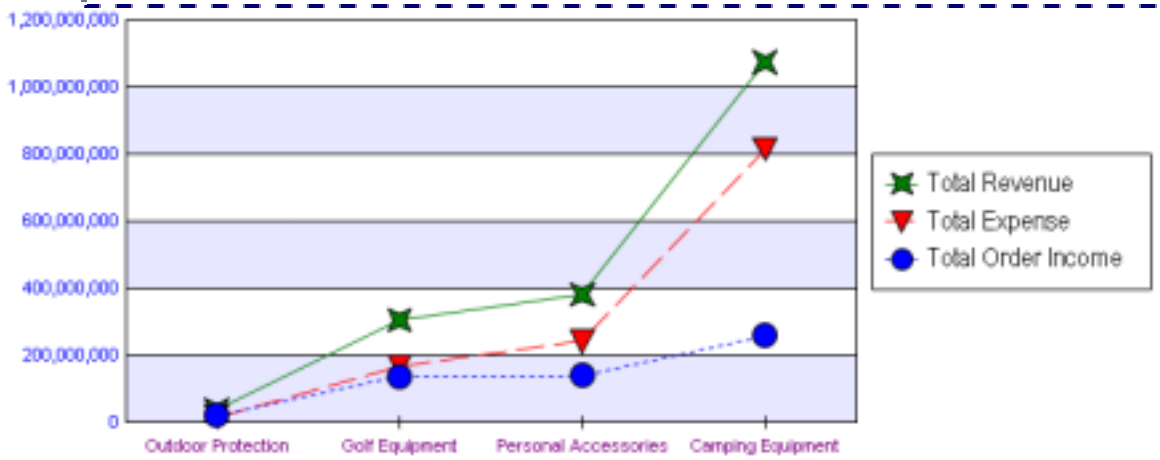
nGreen: 0...255 defines the Green portion of RGB fill color.

nBlue: 0...255 defines the Blue portion of RGB fill color.

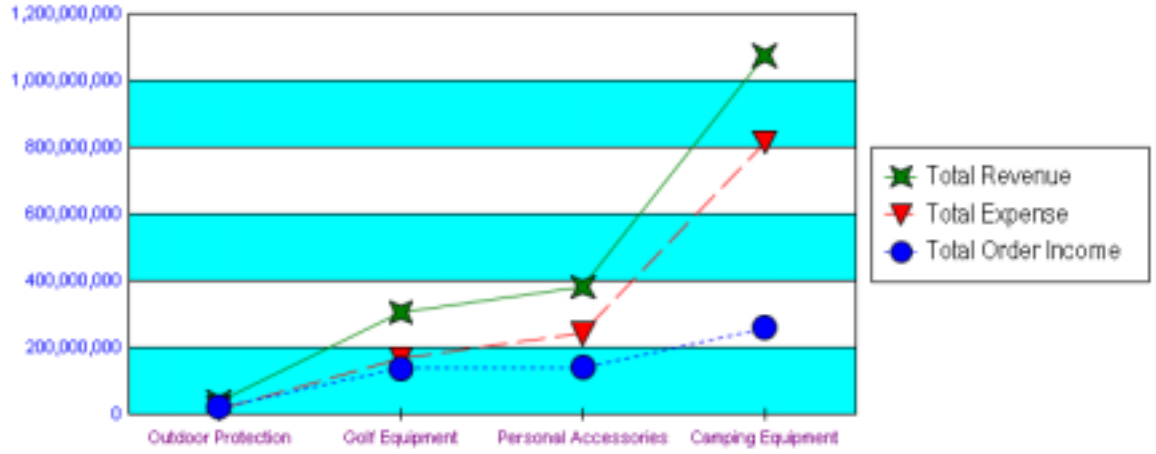
nAlpha: 0...255 selects the transparent level. 255 (the default) selects no transparency. 0 = completely transparent.

EXAMPLE:

```
@ALT_FRAME_FILL_MODE 1
@ALT_FRAME_FILL 0 0 255 25
```




```
@ALT_FRAME_FILL_MODE 1
@ALT_FRAME_FILL 0 255 255 255
```



PERSISTENT:

NO

ALSO SEE:

@ALT_FRAME_FILL_MODE

@ALT_FRAME_FILL_MODE (Alternating Frame Fill Mode)

This macro defines the mode that the @ALT_FRAME_FILL macro will use to color the chart frame with alternating color bands.

SYNTAX:

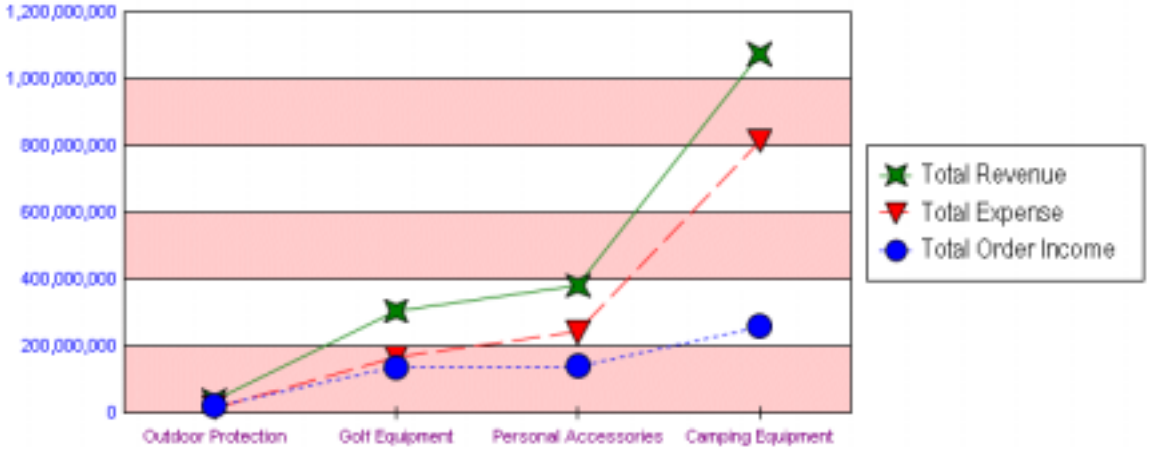
```
@ALT_FRAME_FILL_MODE nMode
```

PARAMETERS:

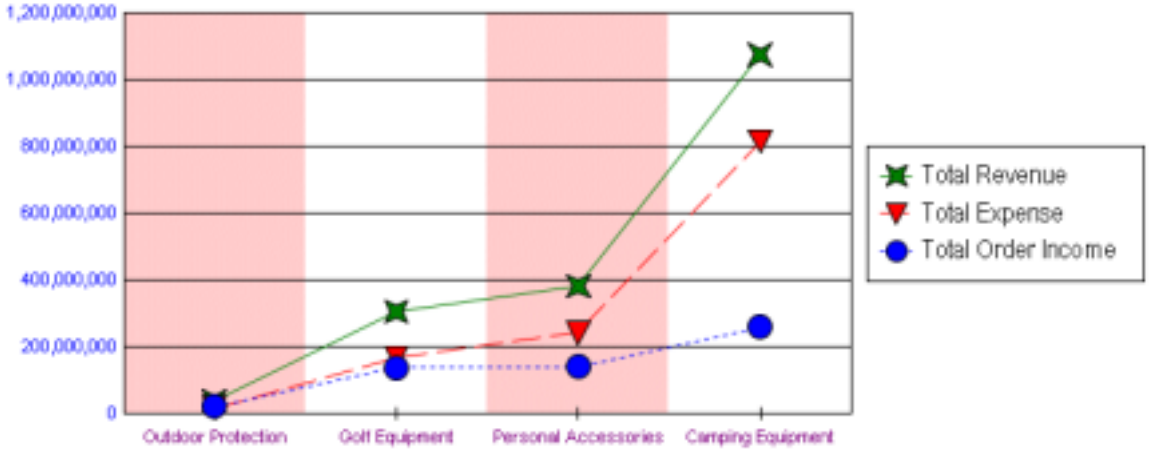
nMode: 0 = Off, 1 = Alternating frame fill on Y-axis, 2 = Alternating frame fill on X-axis.

EXAMPLE:

```
@ALT_FRAME_FILL_MODE 1
@ALT_FRAME_FILL 255 0 0 50
```




```
@ALT_FRAME_FILL_MODE 2
@ALT_FRAME_FILL 255 0 0 50
```



PERSISTENT:

NO

ALSO SEE:

@ALT_FRAME_FILL

@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:

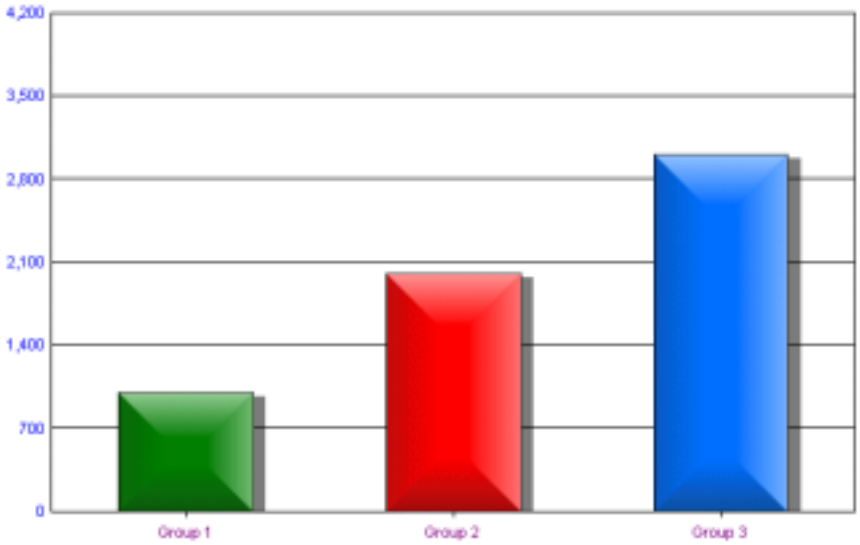
```
@AUTO_COLOR bActivate
```

PARAMETERS:

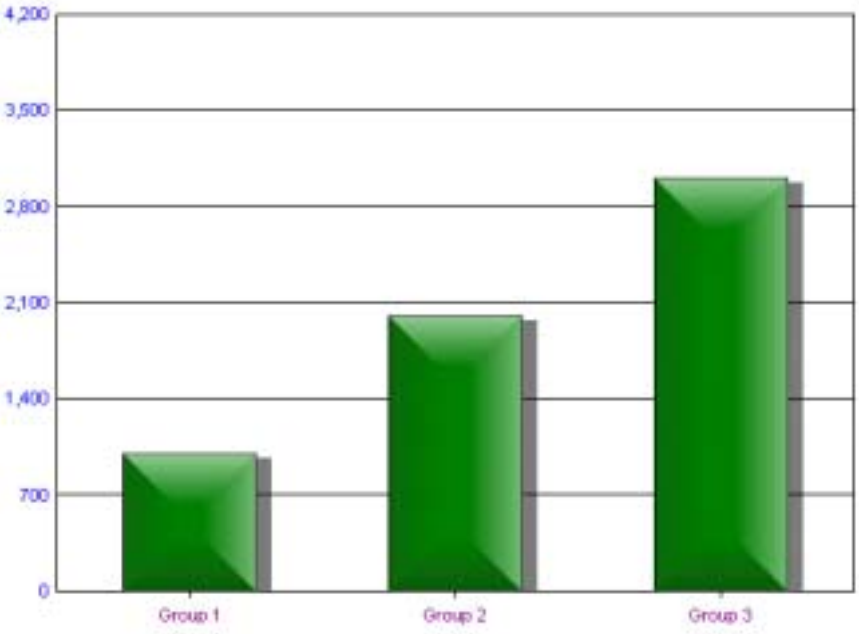
bActivate; 0=Disable Auto-Color Mode, 1=Enable Auto-Color Mode

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
```

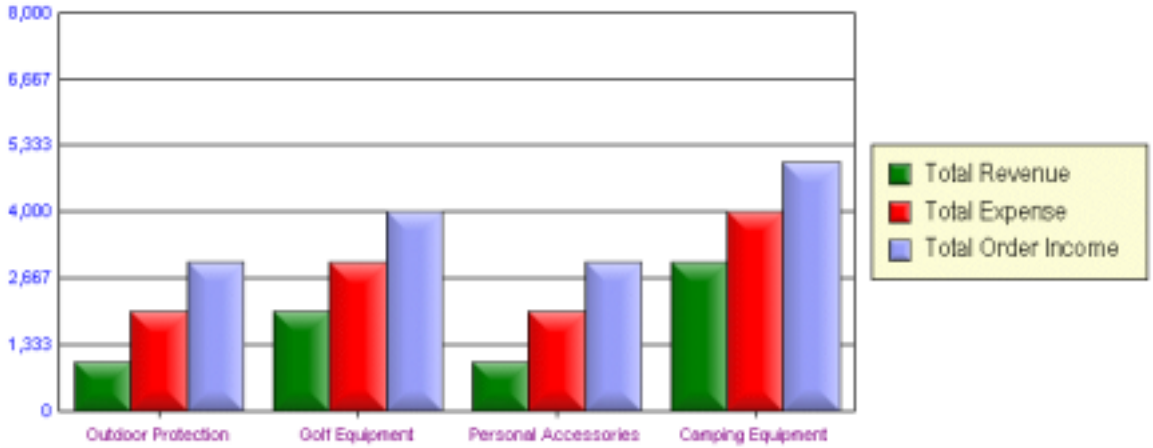
PARAMETERS:

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

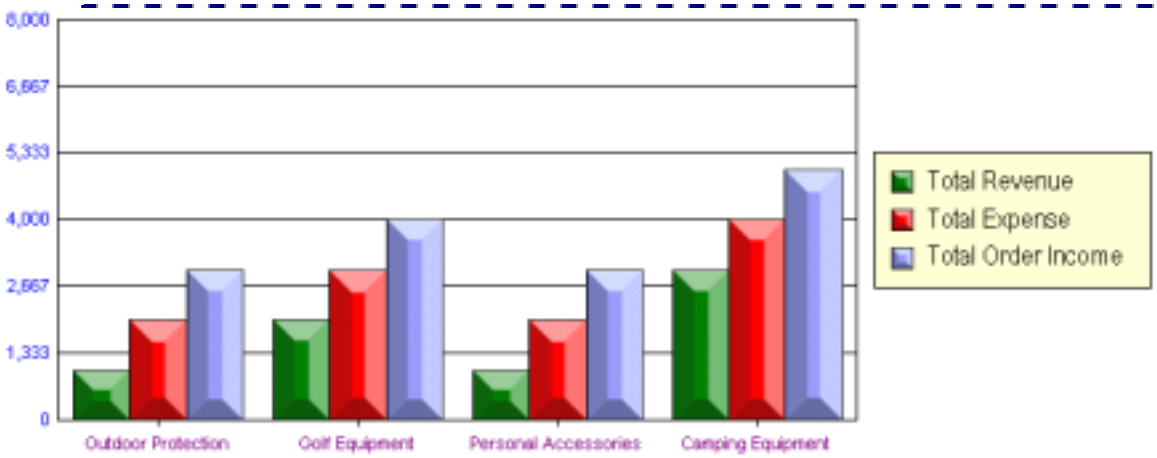
nBevelType; 0...3. 0=None, 1=Smooth Edge, 2=Chisel Edge, 3=Donut, 4=Circular. Bevel types 1 and 2 can only be applied to rectangle and polygon objects. Bevel type 3 can only be applied to ellipses and slices. Bevel type 4 can only be applied to circular markers.

EXAMPLE:

```
@BEVEL 3 1
```




```
@BEVEL 3 2
```



PERSISTENT:

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

@COLOR_MODE (Color Mode)

This macro sets the chart color mode: Color by Series or Color by Group.

SYNTAX:

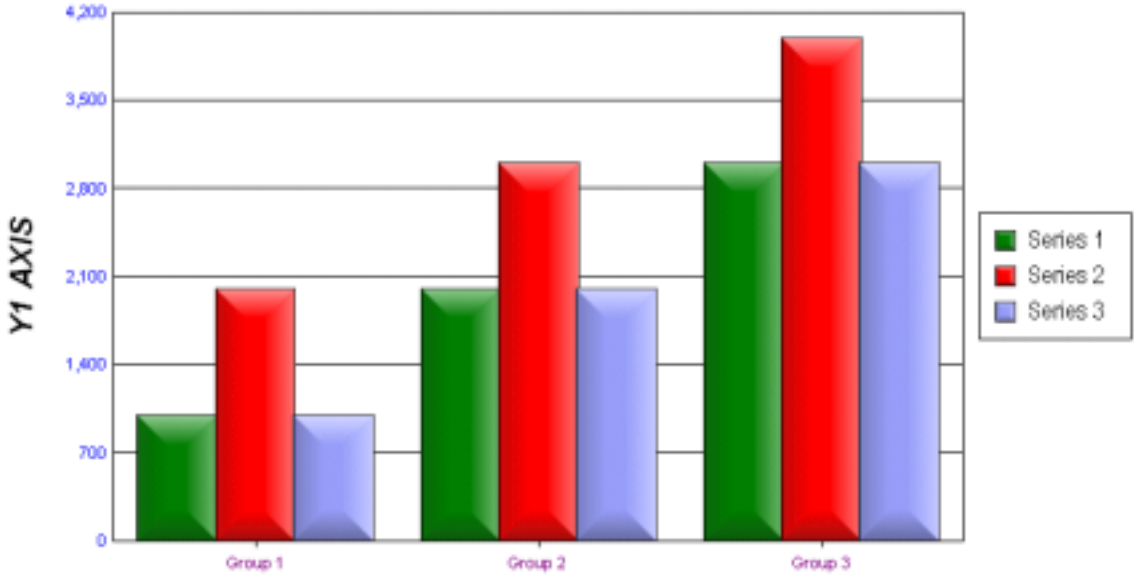
```
@COLOR_MODE nMode
```

PARAMETERS:

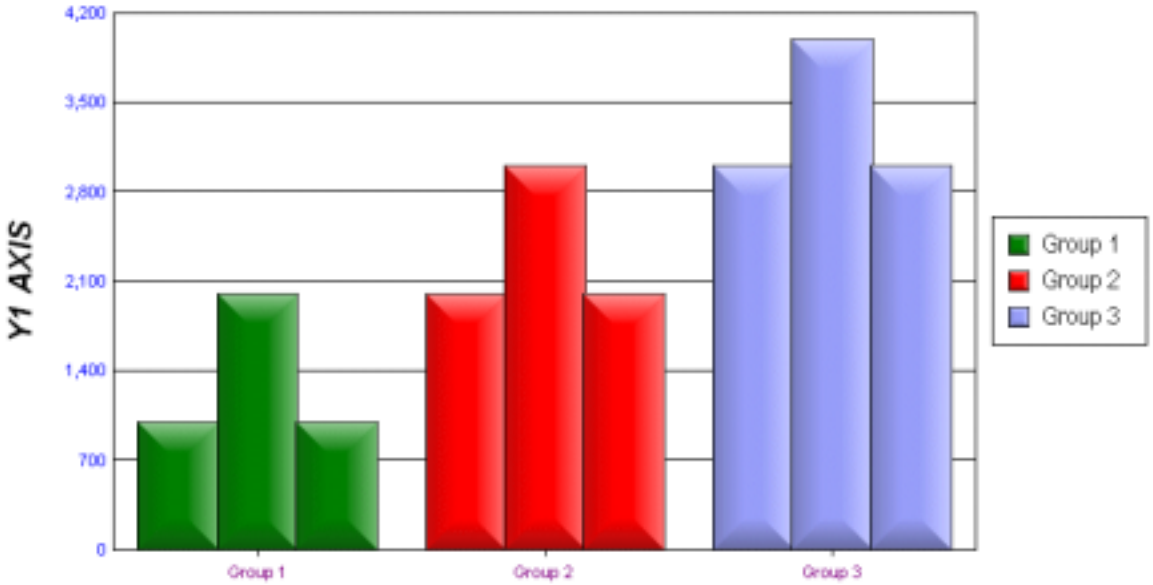
nMode; 0 = Color By Series, 1= Color By Group

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
```

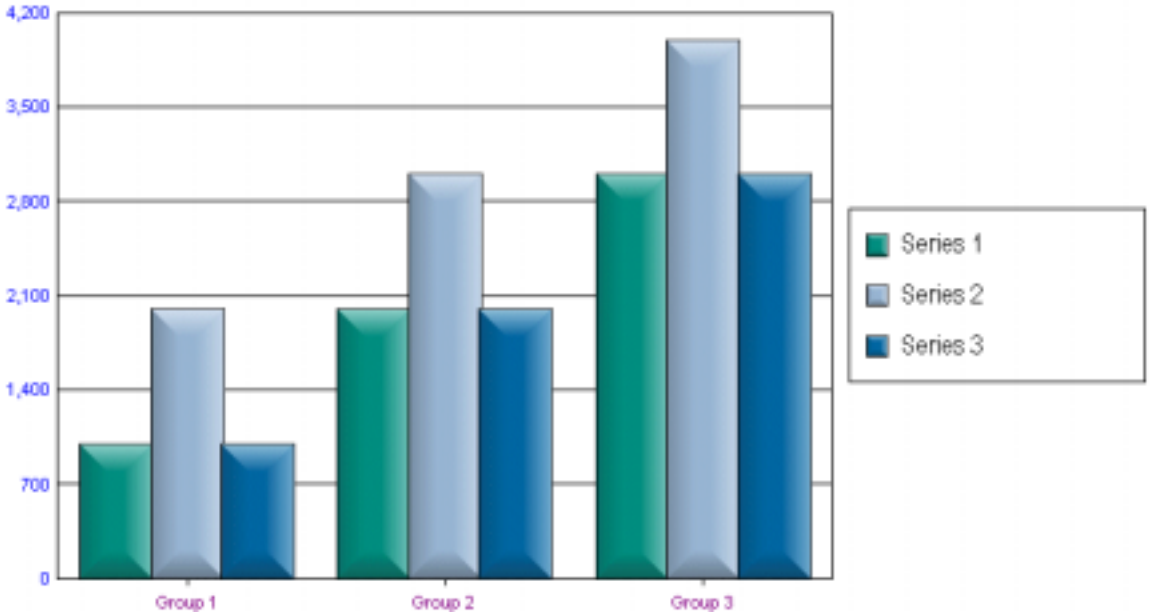
PARAMETERS:

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

nScheme	Color Scheme
0	
1	
2	
3	
4	
5	

EXAMPLE:

```
@COLOR_SCHEME 3
```



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.

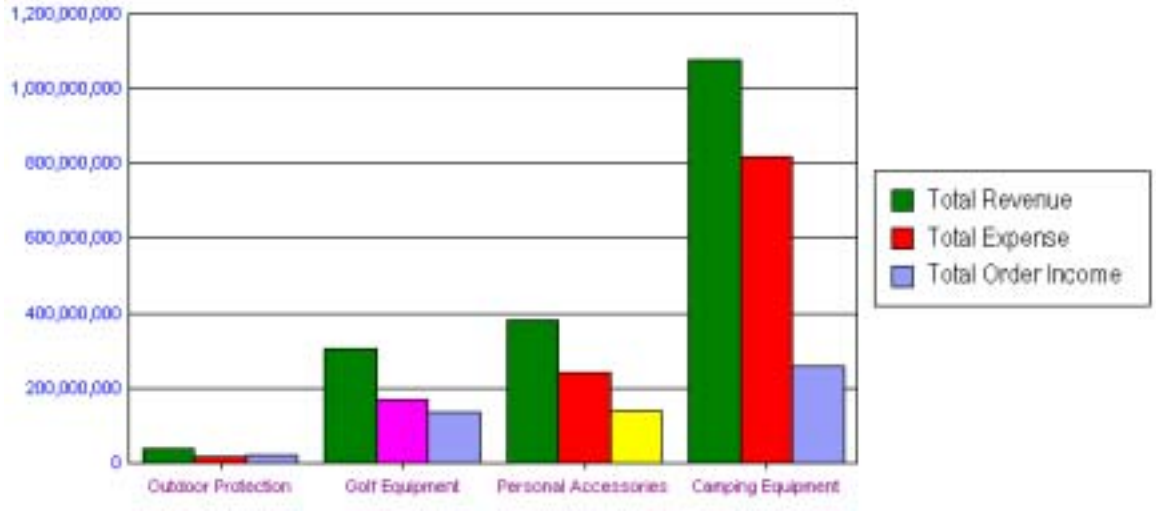
7 = force all series in *nGroup* to the conditional color regardless of *fValue*.

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.

Examples:

```
@COND_COLOR 2 2 0 2000000000 255 255 0
@COND_COLOR2 1 1 2 100000000 255 0 255
```



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:

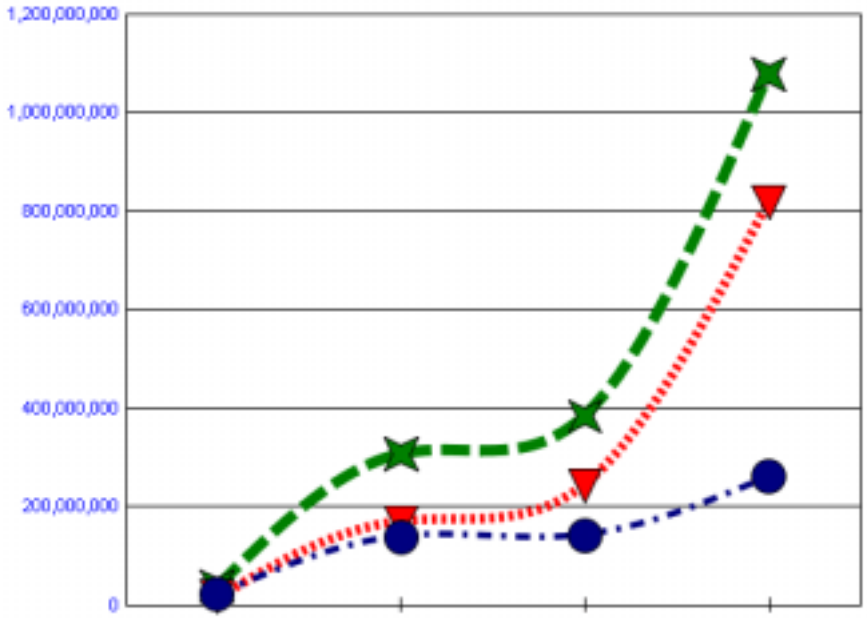
```
@CURVED_LINES bEnable
```

PARAMETERS:

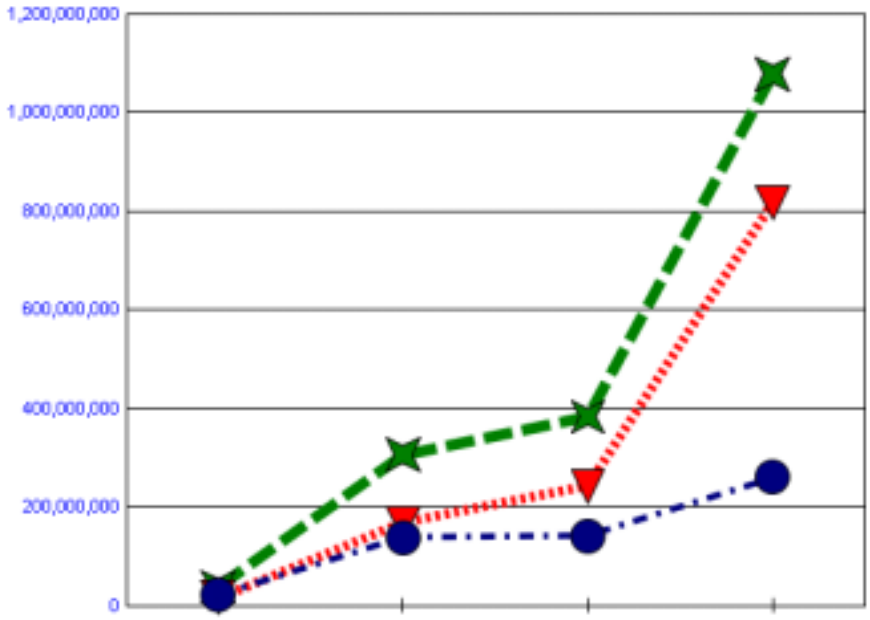
bEnable; 0=Disable Curved Lines, 1=Enable Curved Lines

EXAMPLE:

```
@CURVED_LINES 1
```




```
@CURVED_LINES 0
```



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. Each RGB value can be hex (0xFF0000=red) or decimal (16711680=red).

SYNTAX:

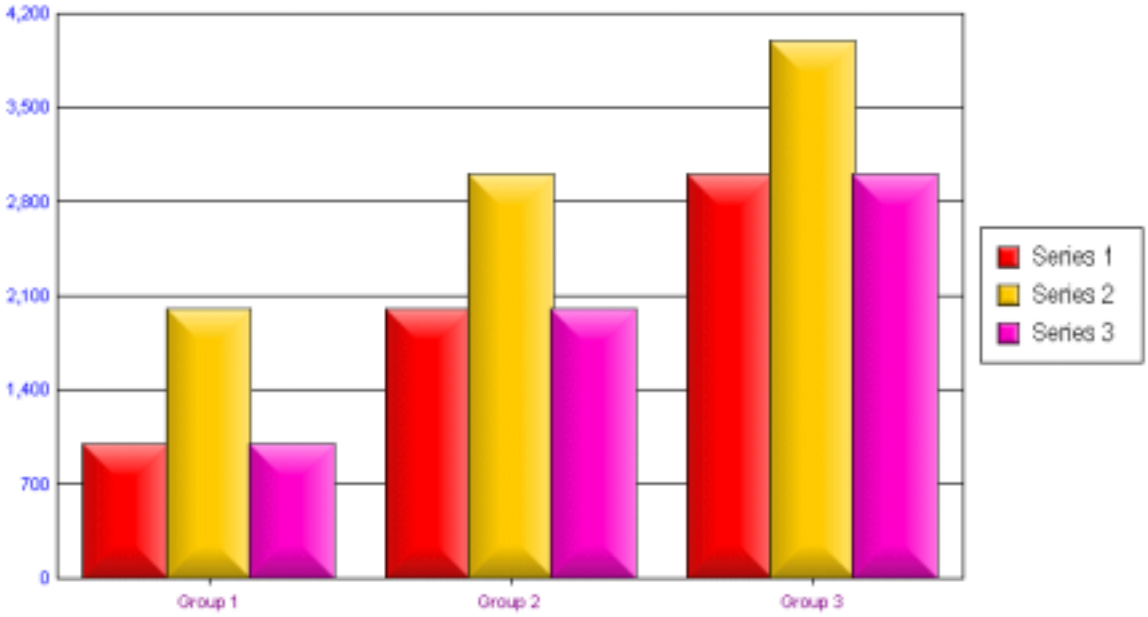
```
@DEFINE_SCHEME rgb0 rgb1 rgb2 rgb3 rgb4 rgb5 rgb6 rgb7
```

PARAMETERS:

rgb0...rgb7; specifies a Red-Green-Blue value in hex or decimal format

EXAMPLE:

```
@DEFINE_SCHEME 0xFF0000 0xFFCC00 0xFF00CC 0xFF0000 0xFFCC00
0xFF00CC 0xFF0000 0xFFCC00
```



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 nRed nGreen nBlue
```

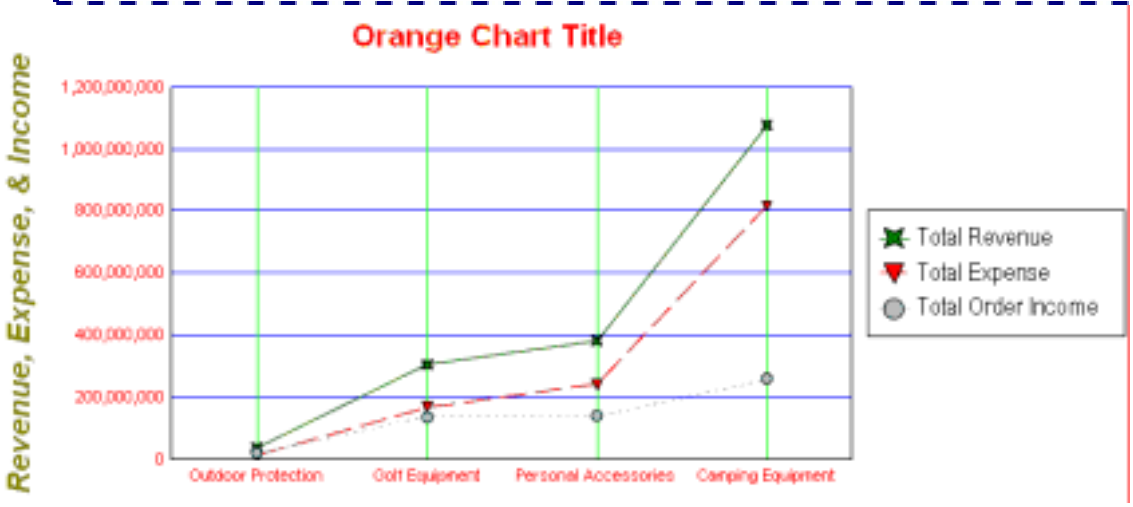
PARAMETERS:

nObject; 0...14 selects one of the following chart objects:
0 = Chart Frame
1 = Legend Frame
2 = Title
3 = Subtitle
4 = Footnote
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

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:

```
@GCOLOR 2 255 125 0
@GCOLOR 5 125 125 0
@GCOLOR 8 255 0 0
@GCOLOR 10 255 0 0
@GCOLOR 11 255 0 0
@GCOLOR 12 0 0 255
@GX 1
@GCOLOR 14 0 255 0
```



PERSISTENT:

YES

@SHADOW (Drop Shadow)

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

SYNTAX:

```
@SHADOW nObject nXOffset nYOffset
```

PARAMETERS:

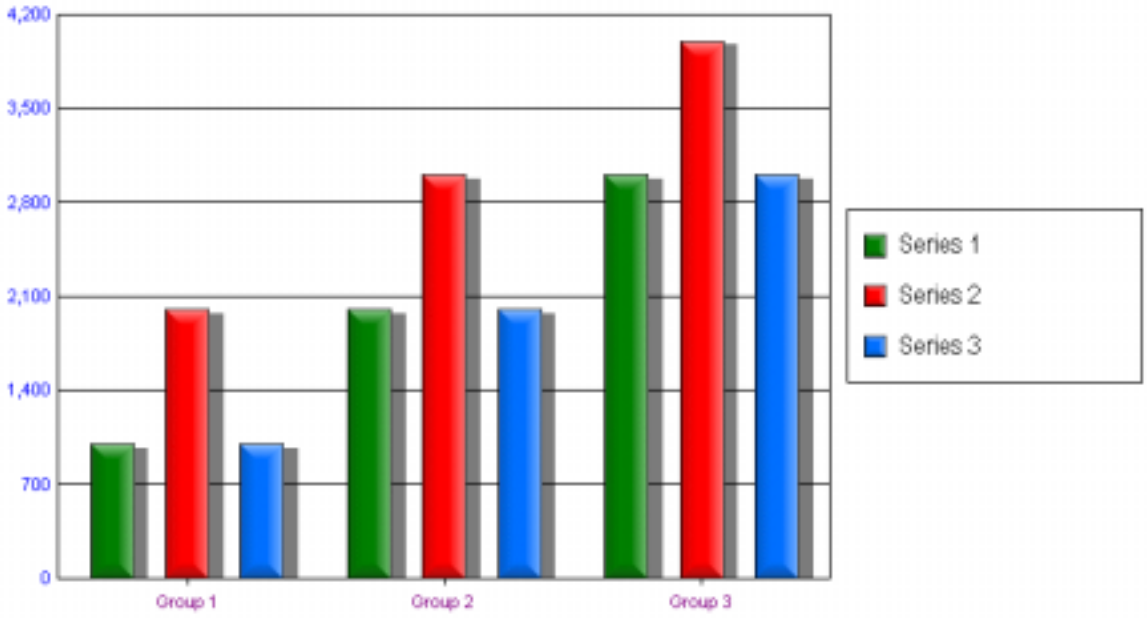
nObject; 0...12 selects one of the following objects:

0 = Chart Frame
1 = Legend Frame
2 = Title
3 = Subtitle
4 = Footnote
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
- @INIT_USERLINES; Initialize all user-defined lines
- @LS; Line Style
- @MEAN; Enable/disable a mean average line across a specified series
- @MOVA; Standard or scientific Moving Average Line to a series
- @SMOOTH_LINE; Connect data points with Smooth or Straight Line segments
- @TRENDLINE; Trend Line across a series
- @TRENDLINE_ALLDATA; Linear Regression Line through all data points
- @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
- @YSZNL; Y1-Axis Line with Label & Value on Left

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

@CX (X-Axis Line with Color)

This macro adds a user-defined line on the X-axis at value *n*. 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 nValue nRed nGreen nBlue
```

PARAMETERS:

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

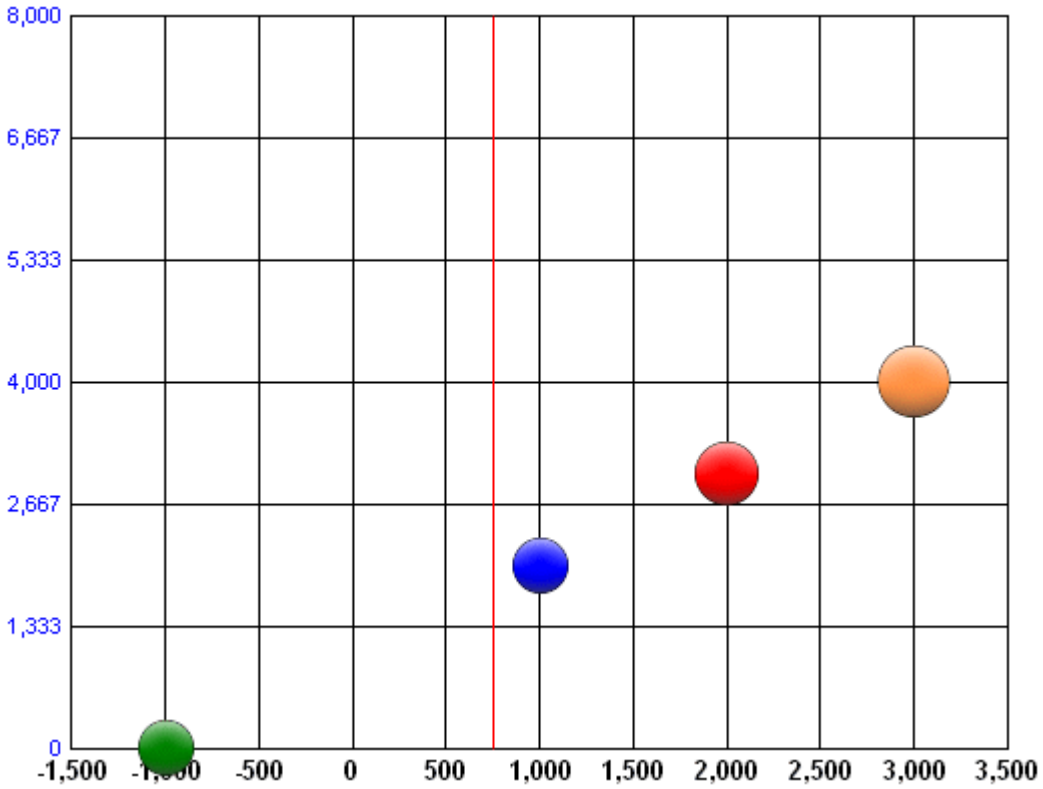
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

EXAMPLE:

```
@CX 750 255 0 0
```



PERSISTENT:

NO

NOTES:

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

@CXY (X/Y Coordinates Line with Color)

This macro adds a user-defined line that starts at location *x1*, *y1* and stops at location *x2*, *y2*. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *x1* and *x2* define the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *x1* and *x2* 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 x1 y1 x2 y2 nRed nGreen nBlue
```

PARAMETERS:

x1; Beginning x-coordinate

y1; Beginning y-coordinate

x2; Ending x-coordinate

y2; Ending y-coordinate

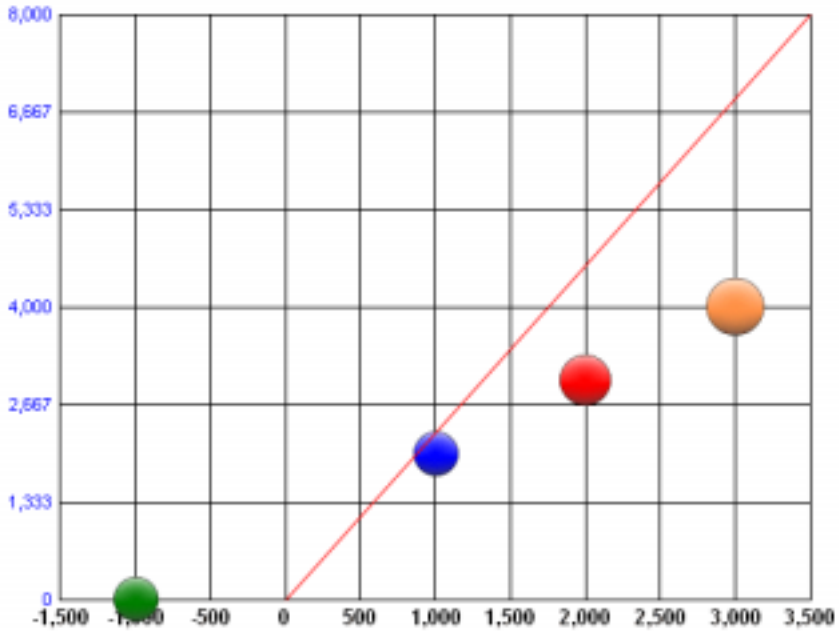
nRed; 0...255 defines the RED portion of the RGB color

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

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

SYNTAX:

```
@CXY 0 0 3500 8000 255 0 0
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with the following macros: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZL, and @YSZNL.

@CY (Y1-Axis Line with Color)

This macro adds a user-defined line on the Y1-axis at value *n*. 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 nValue nRed nGreen nBlue
```

PARAMETERS:

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

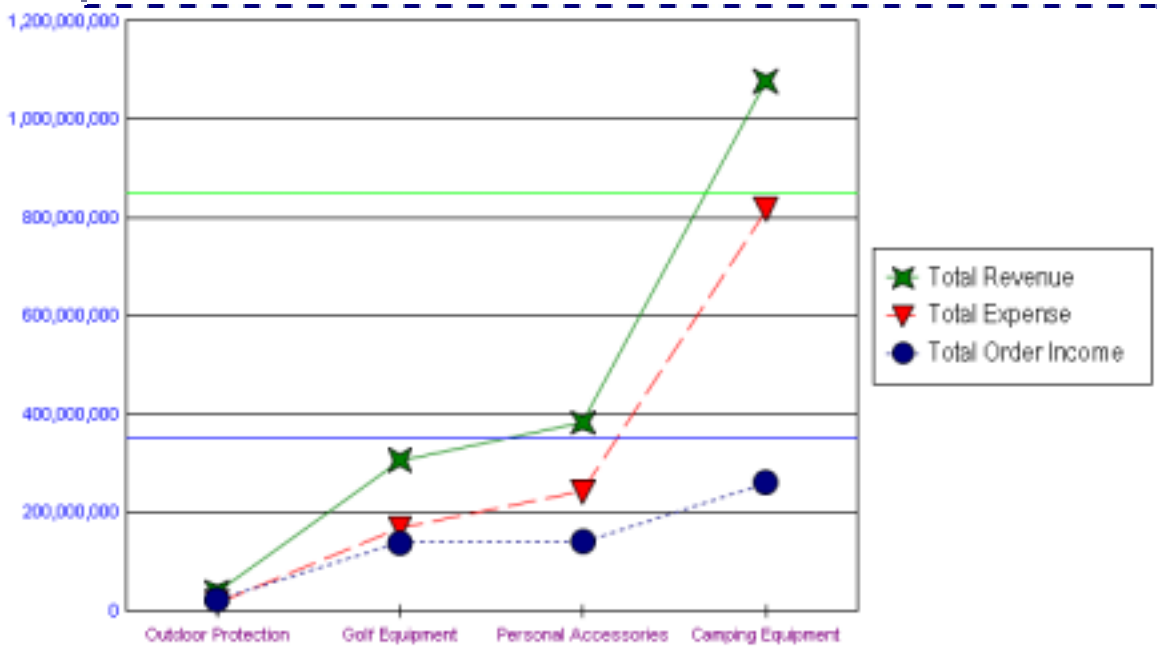
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

EXAMPLE:

```
@CY 350000000 0 0 255
@CY 850000000 0 255 0
```



PERSISTENT:

NO

NOTES:

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

SYNTAX:

```
@INIT_USERLINES
```

PARAMETERS:

None

PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of 20 user-defined lines.

@LS (Line Style)

This macro assigns a thickness and style to a series line.

SYNTAX:

```
@LS nSeries nThickness 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.

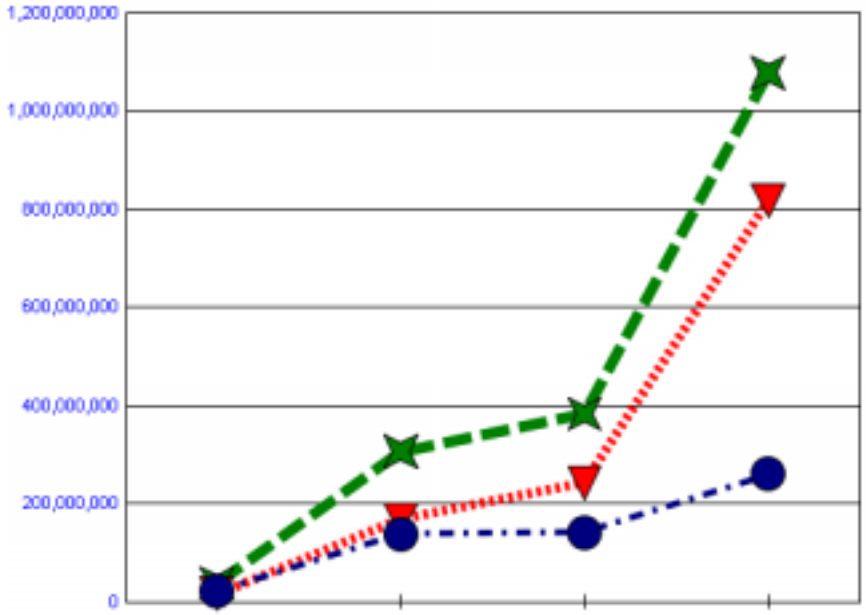
nThickness; 0...1000 selects the thickness of the line.

nStyle; 0...15 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
8 = Long Dot
9 = Dot-Dot-Dot
10 = Dash-Dash-Dot
11 = Dash-Dash-Dot-Dot
12 = Long Dash-Dot
13 = Long Dash-Dot-Dot
14 = Long Dash-Dash-Dot
15 = Long Dash-Dash-Dot-Dot

EXAMPLE:

```
@LS 0 500 1
@LS 1 400 2
@LS 2 300 3
```



PERSISTENT:

YES

@MEAN (Mean Line)

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

SYNTAX:

```
@MEAN nSeries bShow
```

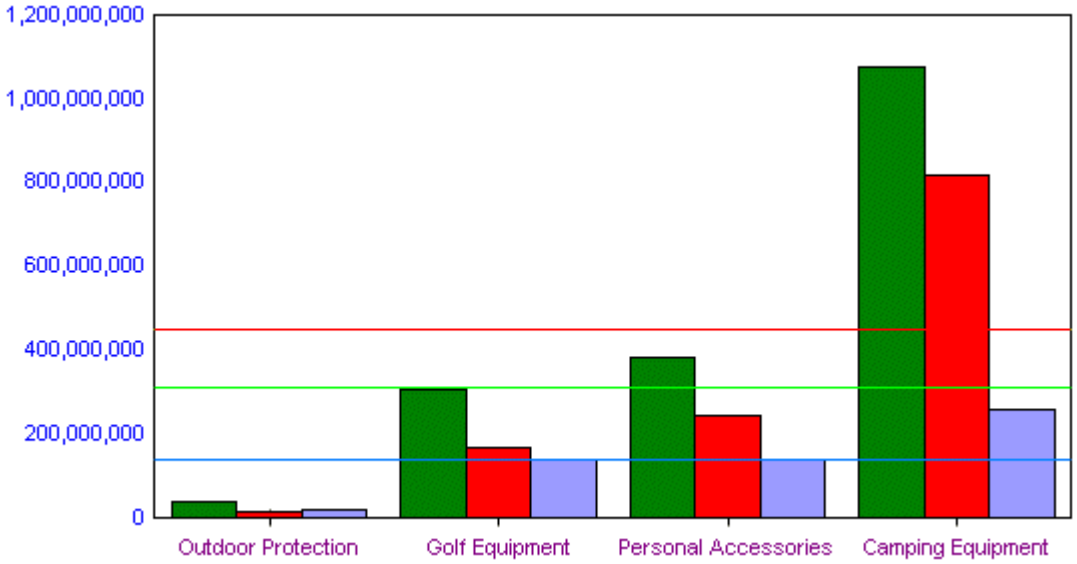
PARAMETERS:

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

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

EXAMPLE:

```
@MEAN 0 1
@MEAN 1 1
@MEAN 2 1
```



PERSISTENT:

YES

@MOVA (Moving Average Line)

This macro applies a scientific or standard moving average line to a specified series. Note that the *bSciMovMode* and *nPeriod* parameters are universal so they will apply to ALL moving average lines being drawn on the chart.

SYNTAX:

```
@MOVA nSeries bShow bSciMovMode nPeriod
```

PARAMETERS:

nSeries; 0...999 series number (0=Series 1)

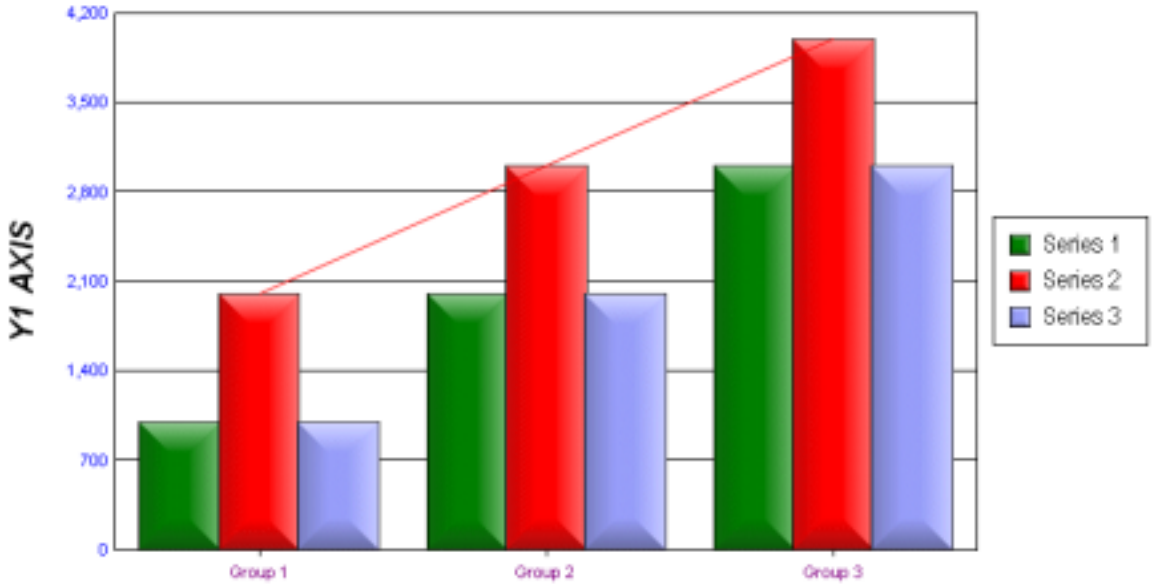
bShow; 1 = Show moving average. 0 = Hide moving average.

bSciMovMode; 1 = Scientific moving average. 0 = Standard moving average

nPeriod; # of periods to use in moving average calculation.

EXAMPLE:

```
@MOVA 1 1 1 1
```



PERSISTENT:

YES

ALSO SEE:

@TRENDLINE to draw a trend line in a chart

@TRENDLINE_ALLDATA to draw a linear regression line through all data points in a chart

@SMOOTH_LINE (Smooth/Straight Lines)

This macro connects data points with a smooth line or straight line segments.

SYNTAX:

```
@SMOOTH_LINE nSeries bOnOff
```

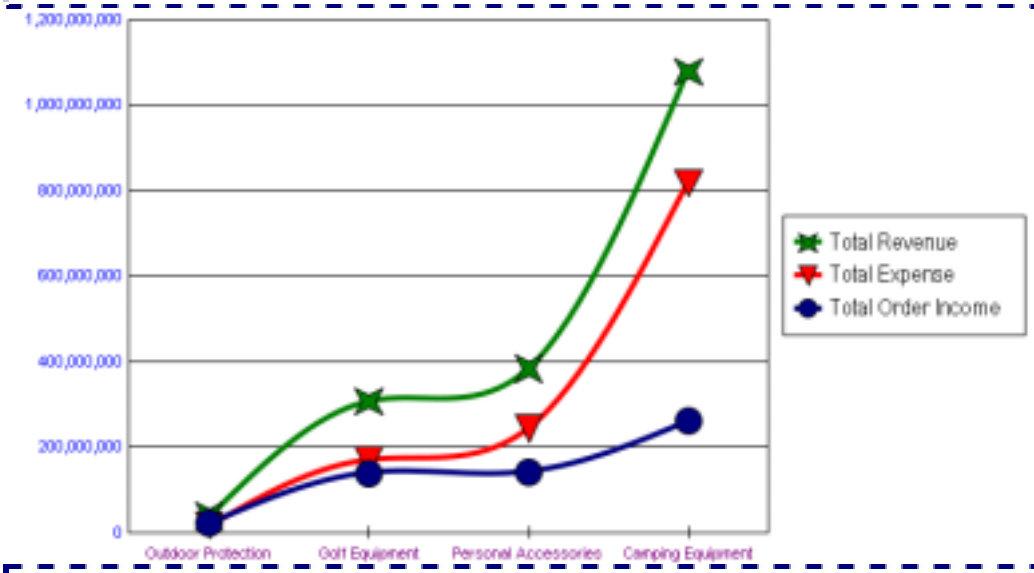
PARAMETERS:

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.

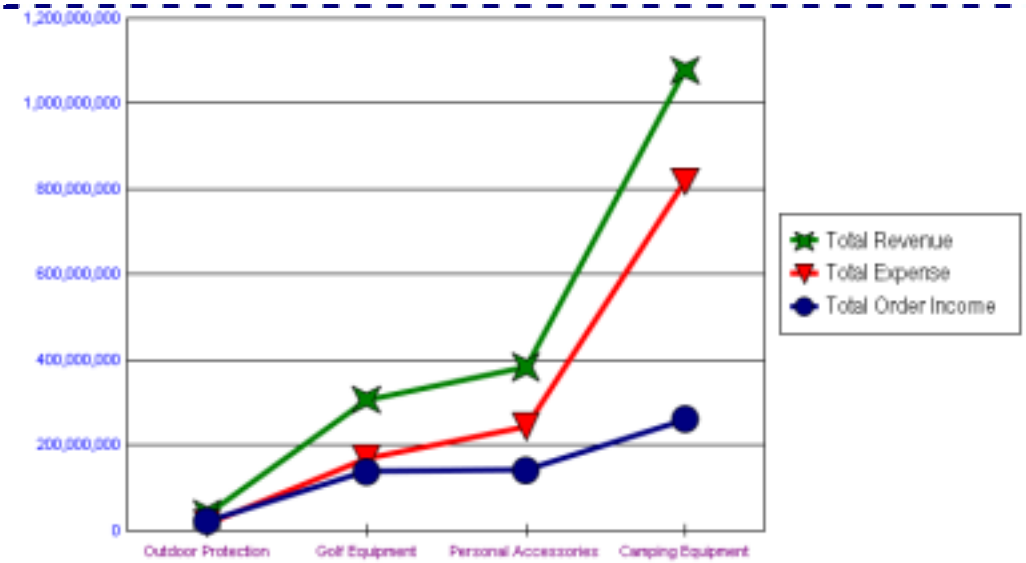
bOnOFF; 1 = connect data markers using a smooth line. 0 = connect data markers with straight-line segments (default).

EXAMPLE:

```
@SZ 50 @LS -1 300 0 @SMOOTH_LINE -1 1
```




```
@SMOOTH_LINE -1 0
```



PERSISTENT:

YES

@TRENDLINE (Trend Line)

This macro draws a trend line across a specified series.

SYNTAX:

```
@TRENDLINE nSeries nValue
```

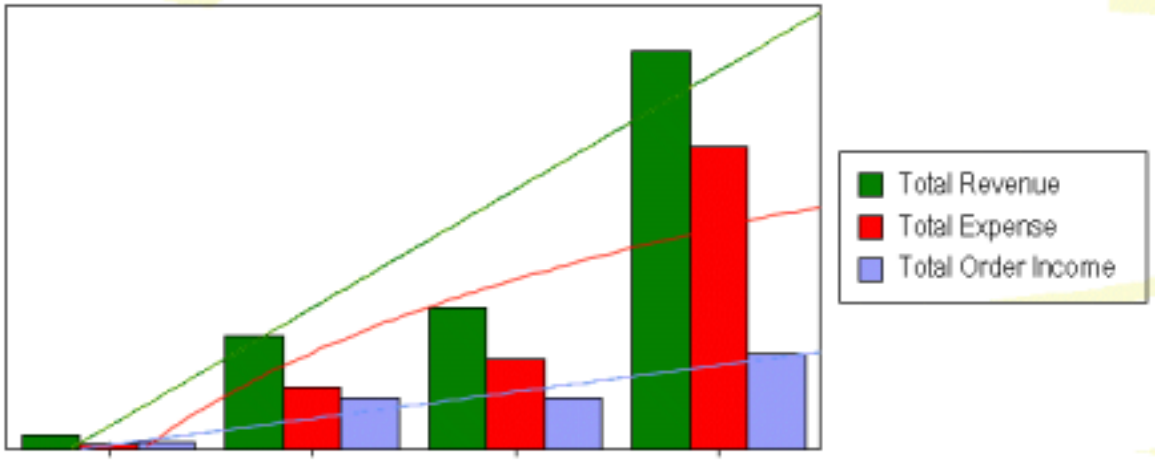
PARAMETERS:

nSeries; Series (0...1000) 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
@TRENDLINE 1 8
@TRENDLINE 2 20
```



PERSISTENT:

YES

ALSO SEE:

@MOVA to draw a moving average line in a chart

@TRENDLINE_ALLDATA to draw a linear regression line through all data points in a chart

@TRENDLINE_ALLDATA (Linear Regression Line)

This macro draws a linear regression line through all data points in a chart.

SYNTAX:

```
@TRENDLINE_ALLDATA bShowLinearLine bShowEquation
```

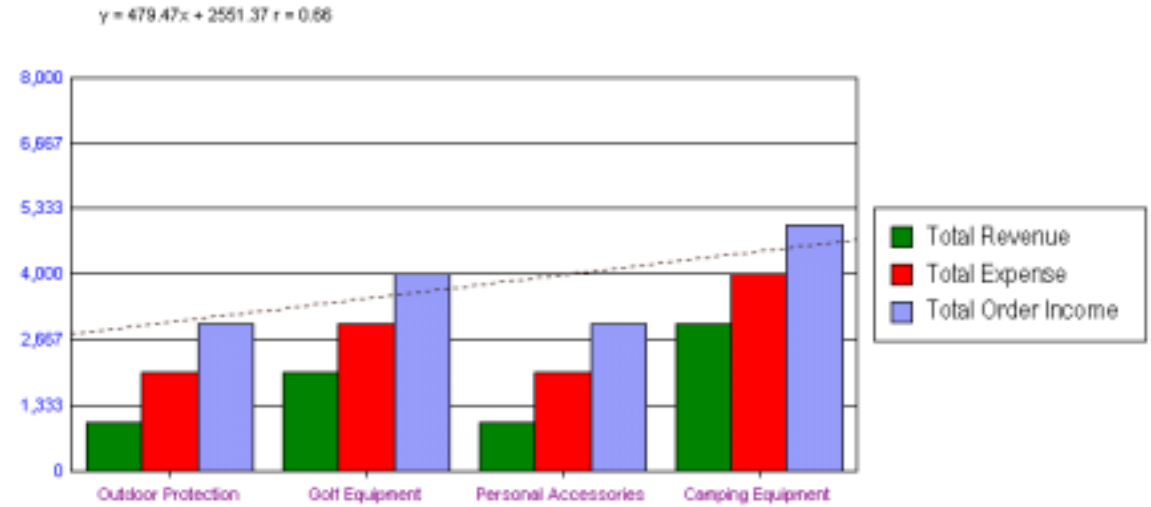
PARAMETERS:

bShowLinearLine; 1=Show Linear Regression Line, 0=Hide Linear Regression Line

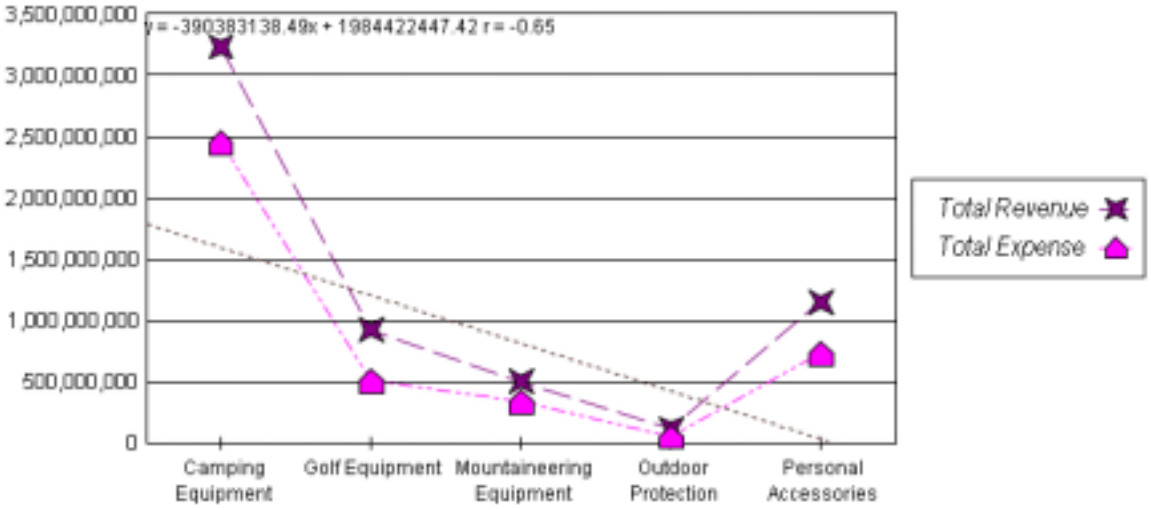
bShowEquation; 1=Show Equation, 1=Hide Equation

EXAMPLE:

```
@TRENDLINE_ALLDATA 1 1
```




```
@TRENDLINE_ALLDATA 1 1
```



PERSISTENT:

YES

ALSO SEE:

@MOVA to draw a moving average line in a chart

@TRENDLINE to draw a trend line

@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.), *n* defines the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *n* 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:

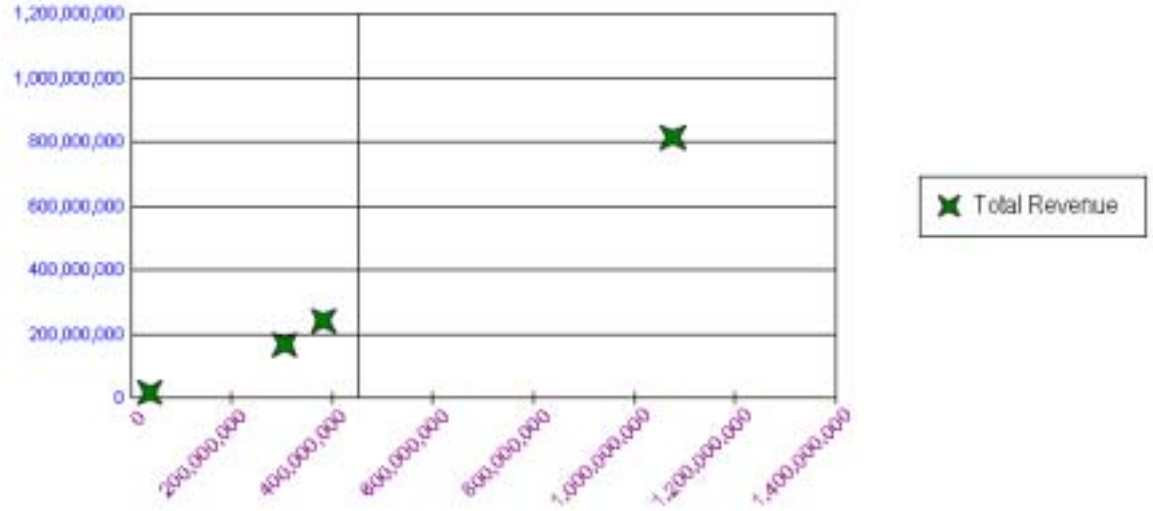
```
@X n
```

PARAMETERS:

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

EXAMPLE:

```
@X 450000000
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 to place the line more accurately (using *groupID*) on Bar/Line/Area and Box Plot charts. The line is drawn horizontally or vertically depending on the chart orientation.

SYNTAX:

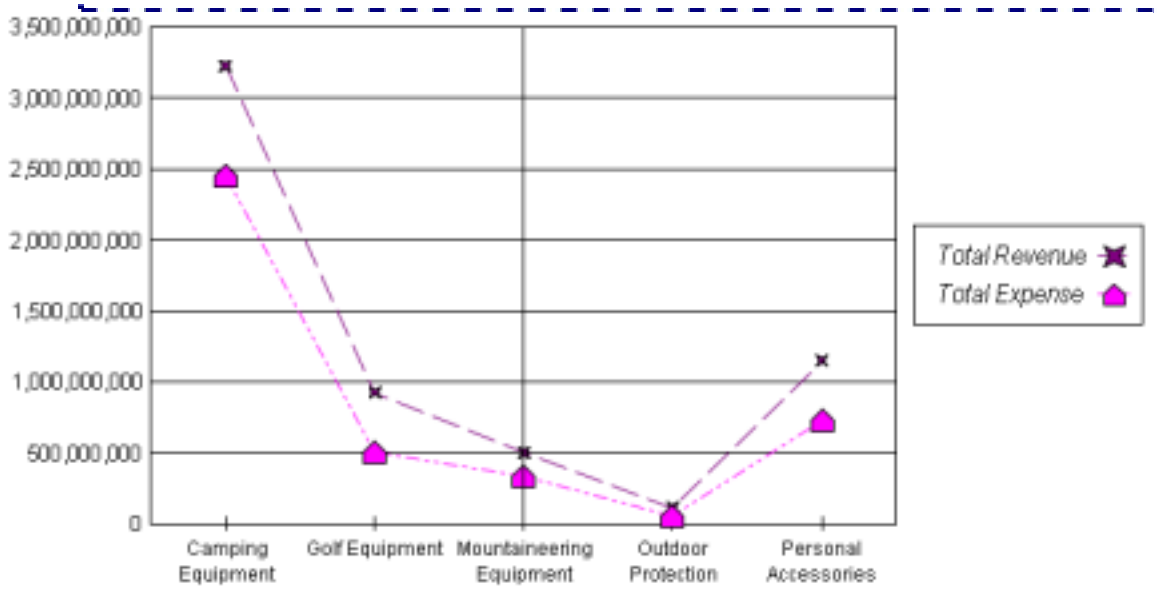
```
@XG groupID
```

PARAMETERS:

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

EXAMPLE:

```
@XG 2
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 nXFixup nYFixup sZLabel
```

PARAMETERS:

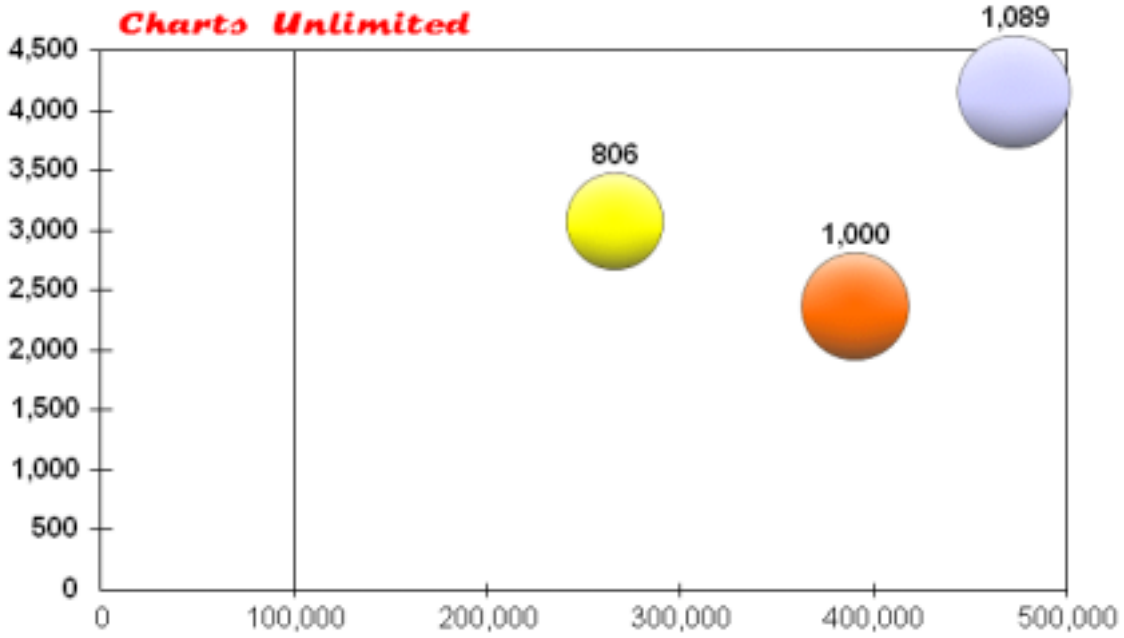
fValue; Value at which to add the user-defined line on the X-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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:

```
@USER_LABEL_FONT 14 255 0 0 255 Splash
@XSZ 100000 0 0 Charts Unlimited
```



PERSISTENT:

NO

ALSO SEE:

@XSZL, @XSZN, @XSZNL

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 nXFixup nYFixup sZLabel
```

PARAMETERS:

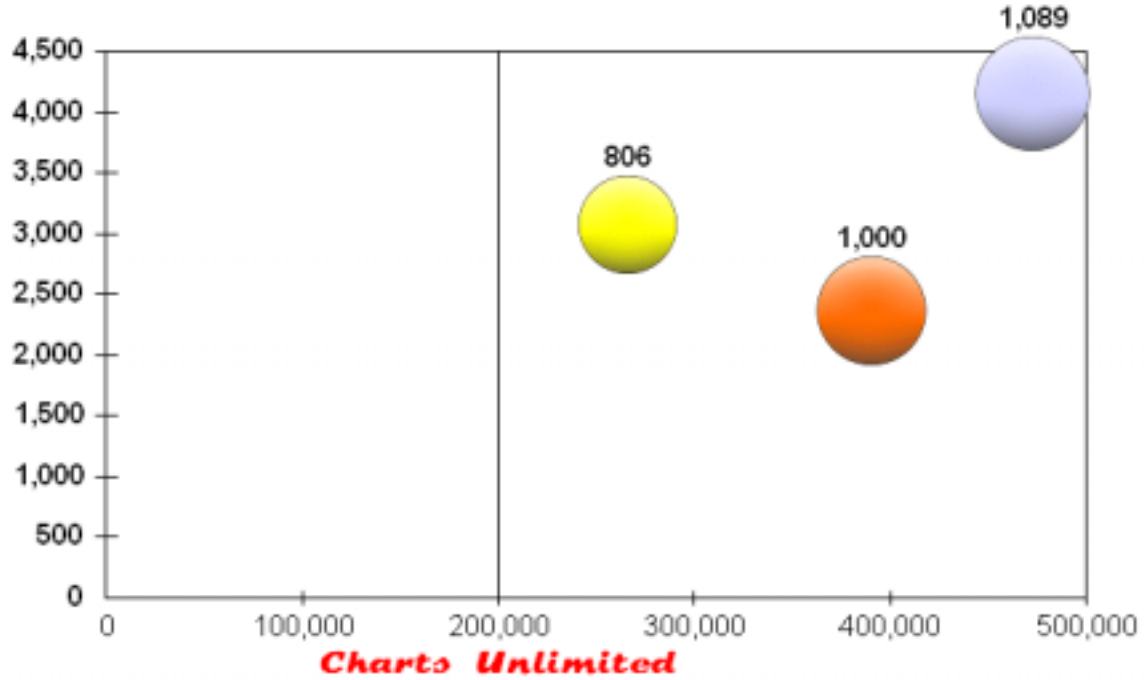
fValue; Value at which to add the user-defined line on the X-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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:

```
@USER_LABEL_FONT 14 255 0 0 255 Splash
@XSZL 200000 0 -2000 Charts Unlimited
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZL, and @YSZNL.

ALSO SEE:

@XSZ, @XSZN, @XSZNL

@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 nXFixup nYFixup sZLabel
```

PARAMETERS:

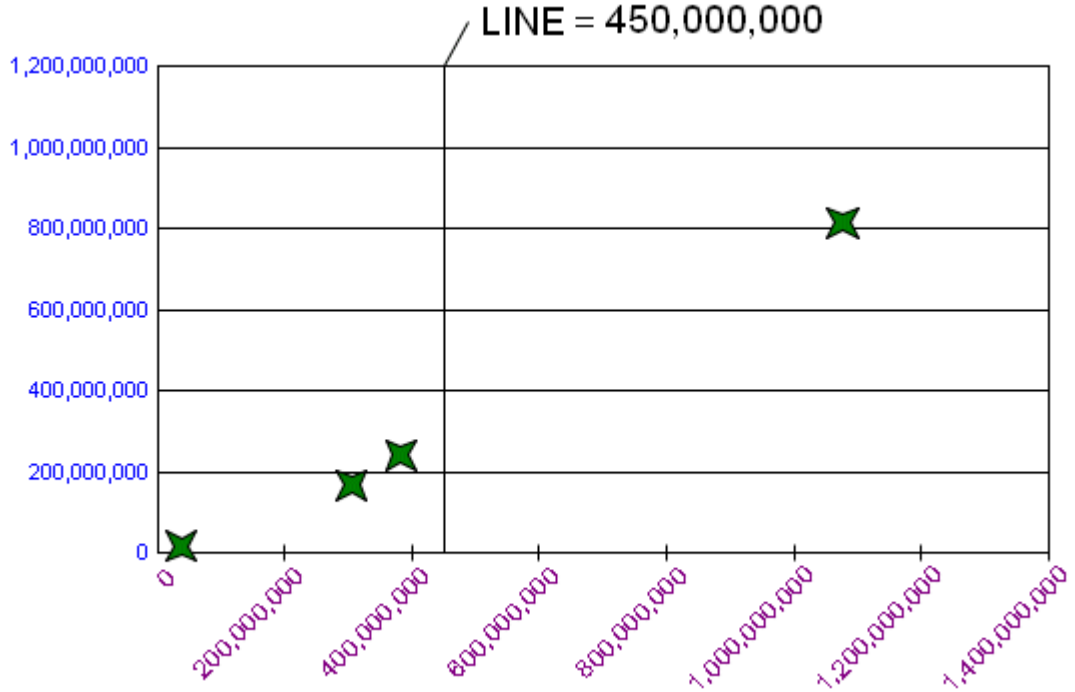
fValue; Value at which to add the user-defined line on the X-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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*.

EXAMPLE:

```
@XSZN 450000000 0 0 LINE =
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZL, and @YSZNL.

ALSO SEE:

@XSZ, @XSZL, @XSZNL

@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 nXFixup nYFixup sZLabel
```

PARAMETERS:

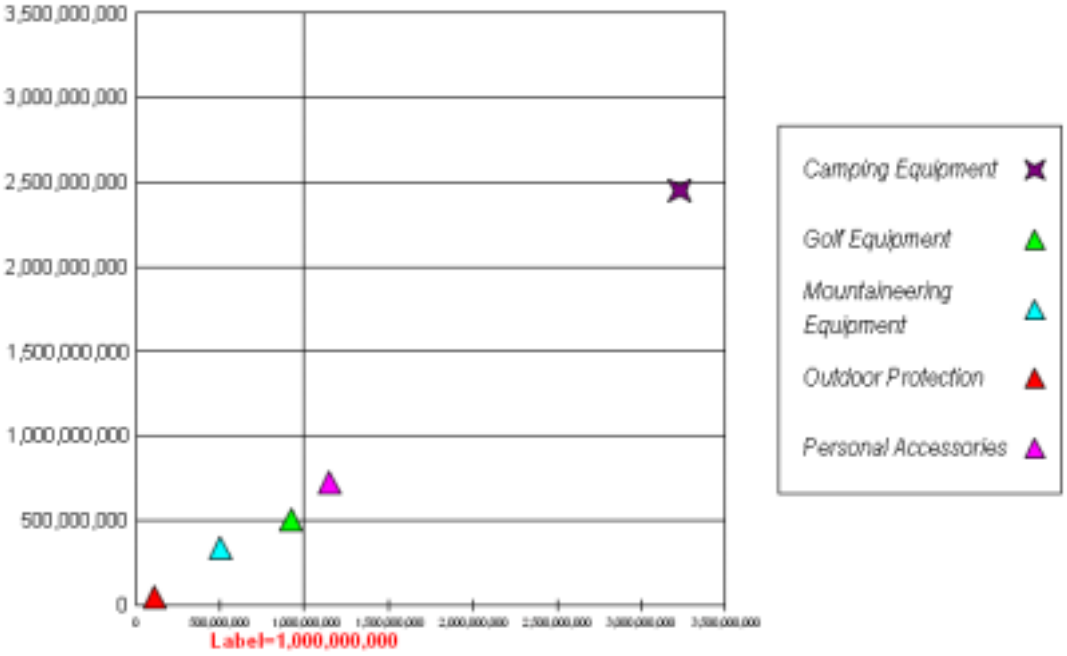
fValue; Value at which to add the user-defined line on the X-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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*.

EXAMPLE:

```
@USER_LABEL_FONT 10 255 0 0 255 Arial
@XSZNL 1000000000 -900 0 Label=
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*). Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZL, and @YSZNL.

ALSO SEE:

@XSZ, @XSZL, @XSZN

@XY (X/Y Coordinates Line)

This macro adds a user-defined line that starts at location *nx1*, *ny1* and stops at location *nx2*, *ny2*. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *nx1* and *nx2* define the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *nx1* and *nx2* 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 nx1 ny1 nx2 ny2
```

PARAMETERS:

nx1; Beginning X-coordinate

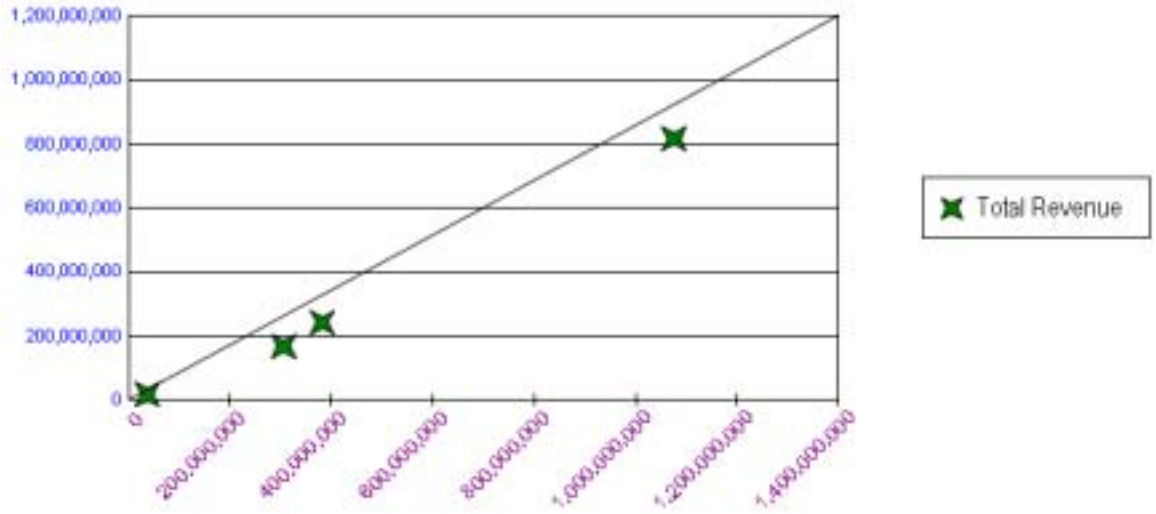
ny1; Beginning Y-coordinate

nx2; Ending X-coordinate

ny2; Ending Y-coordinate

EXAMPLE:

```
@XY 0 0 1400000000 1200000000
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 s1 g1 s2 g2
```

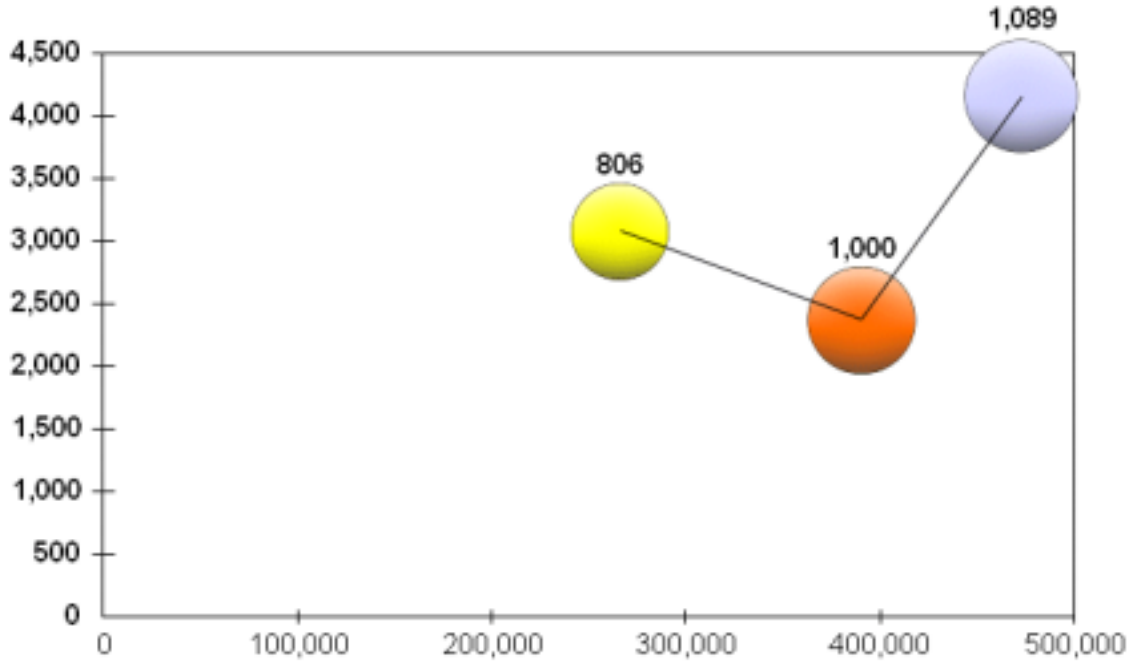
PARAMETERS:

s1, g1; Zero-based series number and group number to start drawing line.

s2, g2; Zero-based series number and group number to stop drawing line.

SYNTAX:

```
@XY_DP2 0 0 1 0
@XY_DP2 1 0 2 0
```



PERSISTENT:

NO

NOTES:

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

@Y (Y1-Axis Line)

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

SYNTAX:

```
@Y n
```

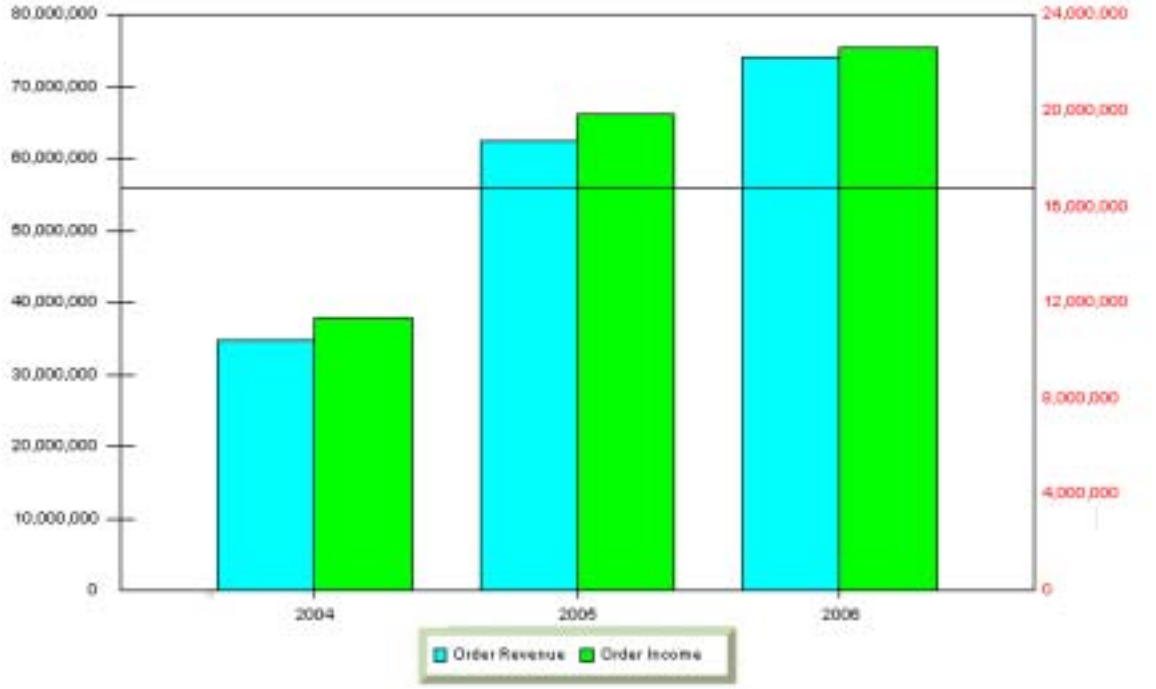
PARAMETERS:

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

EXAMPLE:

This example adds a free line to the chart. The line will be horizontal and start on the Y1-axis at value 4.5.

```
@Y 56000000
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 nXFixup nYFixup sZLabel
```

PARAMETERS:

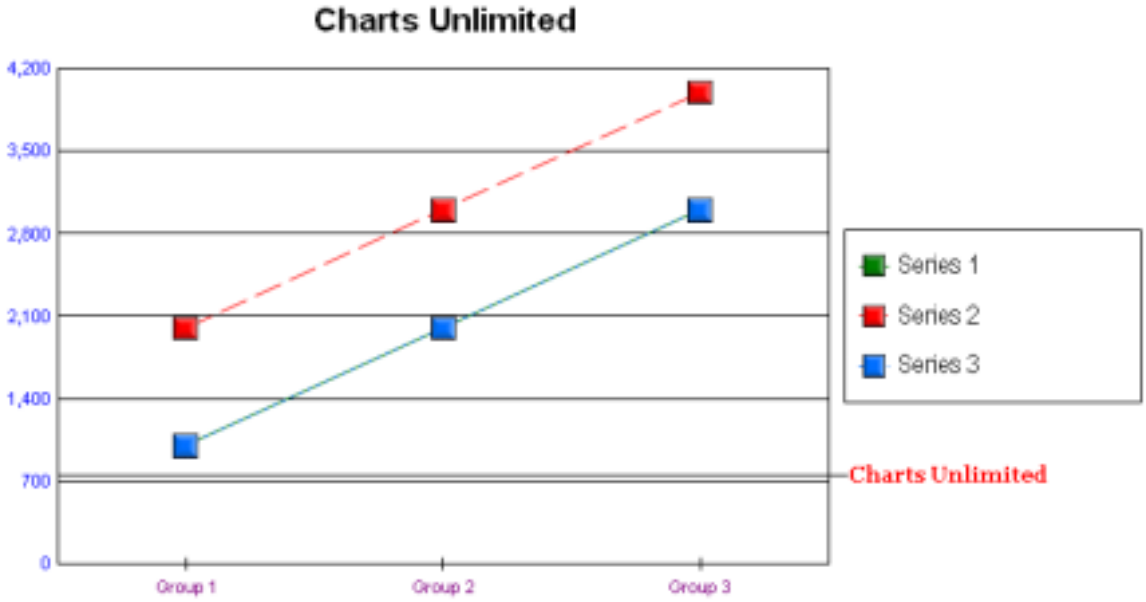
fValue; Value at which to add the user-defined line on the Y1-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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:

```
@USER_LABEL_FONT 12 255 0 0 255 Georgia
@YSZ 750 0 0 Charts Unlimited
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 nXFixup nYFixup sZLabel
```

PARAMETERS:

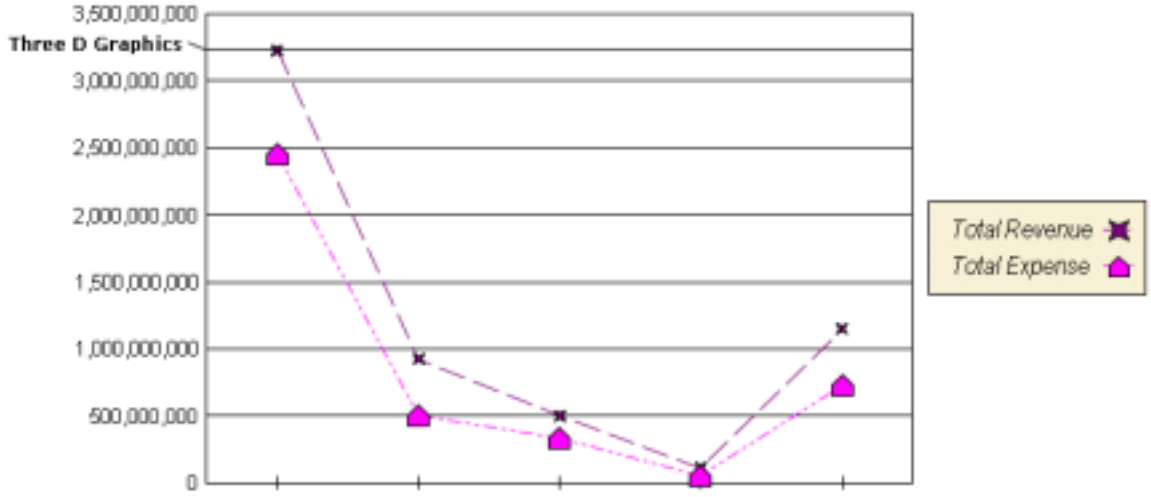
fValue; Value at which to add the user-defined line on the Y1-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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:

```
@YSZL 3250000000 0 0 Three D Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @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 nXFixup nYFixup sZLabel
```

PARAMETERS:

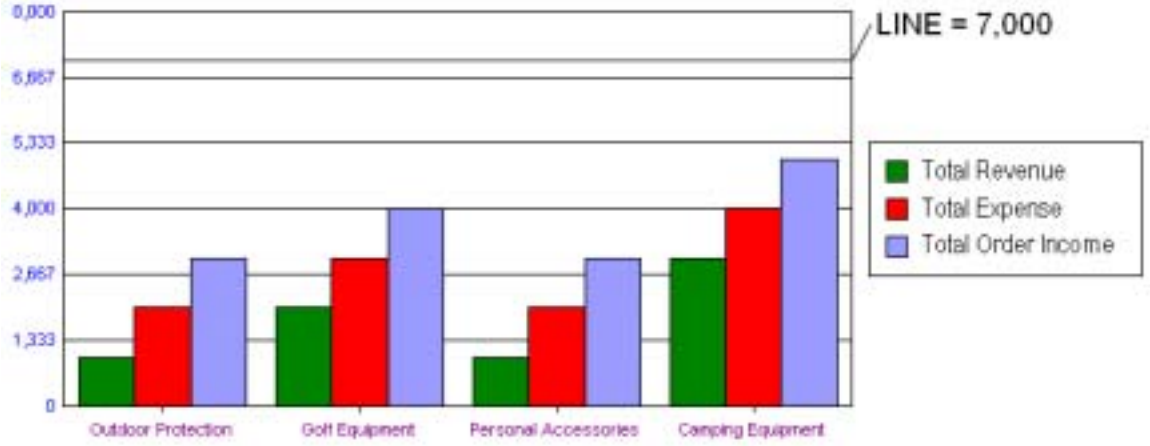
fValue; Value at which to add the user-defined line on the Y1-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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*.

EXAMPLE:

```
@YSZN 7000 0 0 LINE =
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZL, and @YSZNL.

ALSO SEE:

@YSZ, @YSZL, & @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 nXFixup nYFixup sZLabel
```

PARAMETERS:

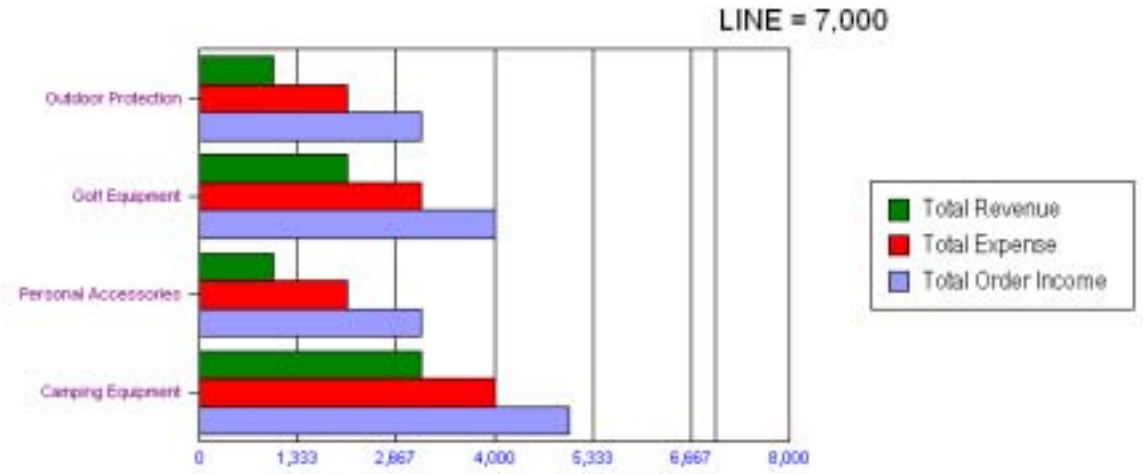
fValue; Value at which to add the user-defined line on the Y1-axis

nXFixup, nYFixup; -16000...16000 adjusts the label position 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*.

EXAMPLE:

```
@YSZNL 7000 0 0 LINE =
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*sZLabel*).
- Charts Unlimited supports a maximum of 20 user-defined lines. User-defined lines are drawn with: @CX, @CXY, @CY, @X, @XG, @XSZ, @XSZL, @XSZN, @XSZNL, @XY, @XY_DP2, @Y, @YSZ, @YSZN, @YSZL, and @YSZNL.

ALSO SEE:

@YSZ, @YSZL, & @YSZN

Section 10: User-Defined Objects

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_MARKER2; Draw a User-Defined Marker at X/Y coordinates with Value
- @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. The *nRed/NGreen/nBlue*, *nThickness*, and *nStyle* parameters define the color, thickness, and line style of the circle. Set *fStartX/ fStopX* to zero to select the lower left corner of the chart frame. The optional phrase (*szPhrase*) is drawn in the middle of the circle.

SYNTAX:

```
@USER_CIRCLE fStartX fStopX fStartY fStopY nRed nGreen nBlue
nStyle nThickness 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/*nGreen*/*nBlue*; 0...255: Define the circle color.

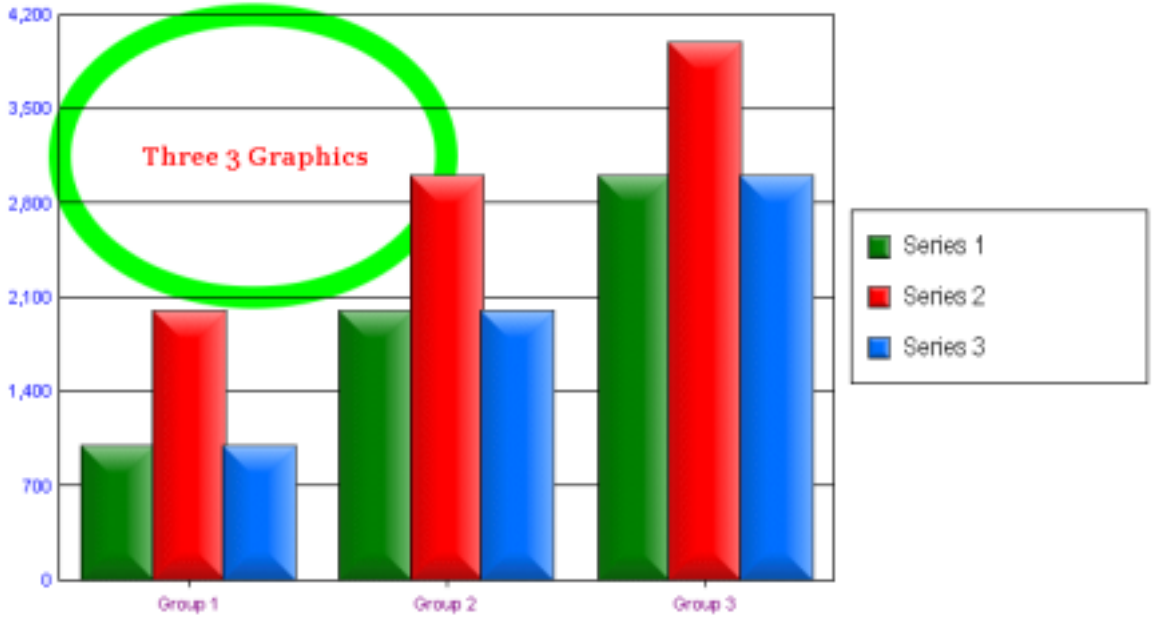
nStyle; 0...15 selects the circle's outline style (0=Solid, 1=Dashed, 2=Dotted, 3=Dot-Dash, 4=Dash-Dot-Dot, 5=Medium Dash, 6=Short Dash, 7=Long Dash, 8=Long Dot, 9=Dot-Dot-Dot, 10=Dash-Dash-Dot, 11=Dash-Dash-Dot-Dot, 12=Long Dash-Dot, 13=Long Dash-Dot-Dot, 14=Long Dash-Dash-Dot, 15=Long Dash-Dash-Dot-Dot)

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

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

EXAMPLE:

```
@USER_LABEL_FONT 12 255 0 0 255 Georgia
@USER_CIRCLE 0.0 0.5 1.0 0.5 0 255 0 0 1000 Three D Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).

- Charts Unlimited 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 that is drawn above the plot area. The *nRed/NGreen/nBlue*, *nThickness*, and *nStyle* parameters define the color, thickness, and line style of the circle. Set *fStartX/ fStopX* to zero to select the lower left corner of the chart frame. The optional phrase (*szPhrase*) is drawn in the middle of the circle.

SYNTAX:

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

PARAMETERS:

fStartX/fStopX; 0.0...1.1 X-Axis start/stop location (0,0=Lower/Left corner of chart frame)

fStartY/fStopY; 0.0...1.1 Y-Axis start/stop location (1.1=Upper/Right corner of chart frame)

nRed/*nGreen*/*nBlue*; 0...255: Define the circle color.

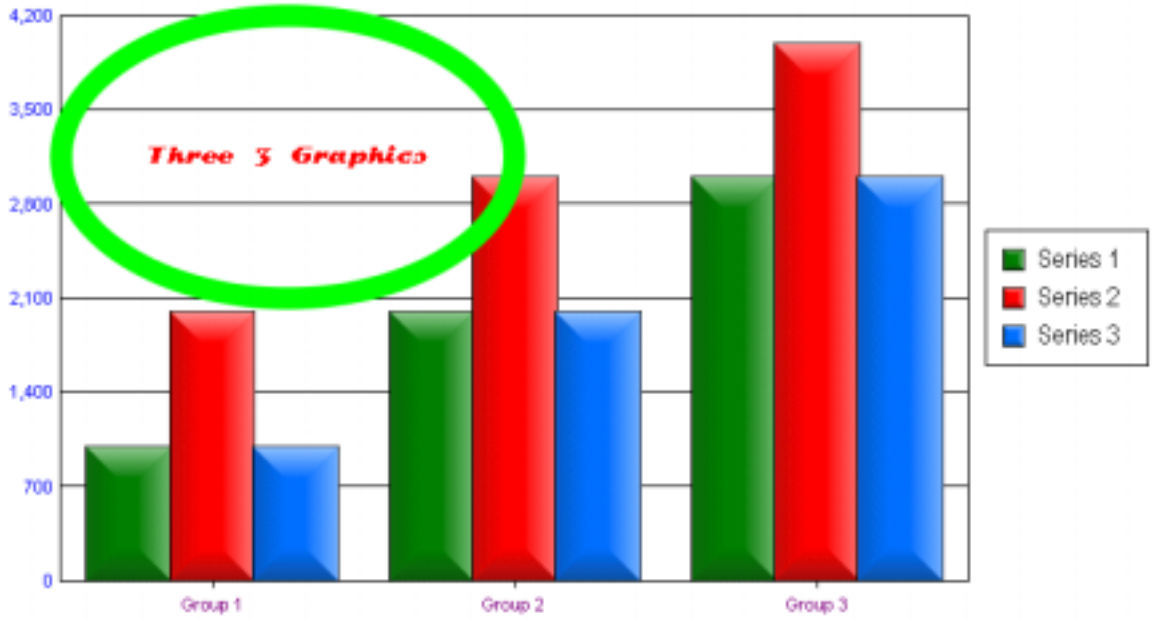
nStyle; 0...15 selects the circle outline style (0=Solid, 1=Dashed, 2=Dotted, 3=Dot-Dash, 4=Dash-Dot-Dot, 5=Medium Dash, 6=Short Dash, 7=Long Dash, 8=Long Dot, 9=Dot-Dot-Dot, 10=Dash-Dash-Dot, 11=Dash-Dash-Dot-Dot, 12=Long Dash-Dot, 13=Long Dash-Dot-Dot, 14=Long Dash-Dash-Dot, 15=Long Dash-Dash-Dot-Dot)

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

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

EXAMPLE:

```
@USER_LABEL_FONT 12 255 0 0 255 Splash
@USER_CIRCLE_ABOVE 0.0 0.5 1.0 0.5 0 255 0 0 1000 Three D Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).
- Charts Unlimited 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 location

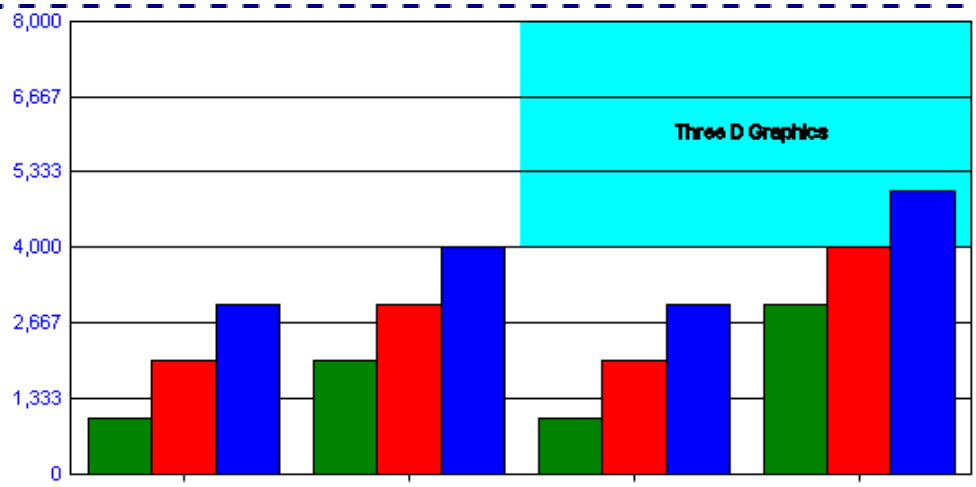
fStartY/*fStopY*; 0.0...1.1 Y-Axis start/stop location

nRed/*nGreen*/*nBlue*; 0...255: Define the rectangle fill color

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

EXAMPLE:

```
@USER_FILL 0.5 1.0 0.5 1.0 0 255 255 Three D Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).
- Charts Unlimited 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
```

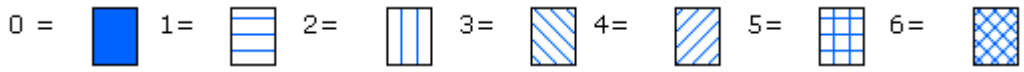
PARAMETERS:

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

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

nRed/*nGreen*/*nBlue*; 0...255: Defines the pattern color.

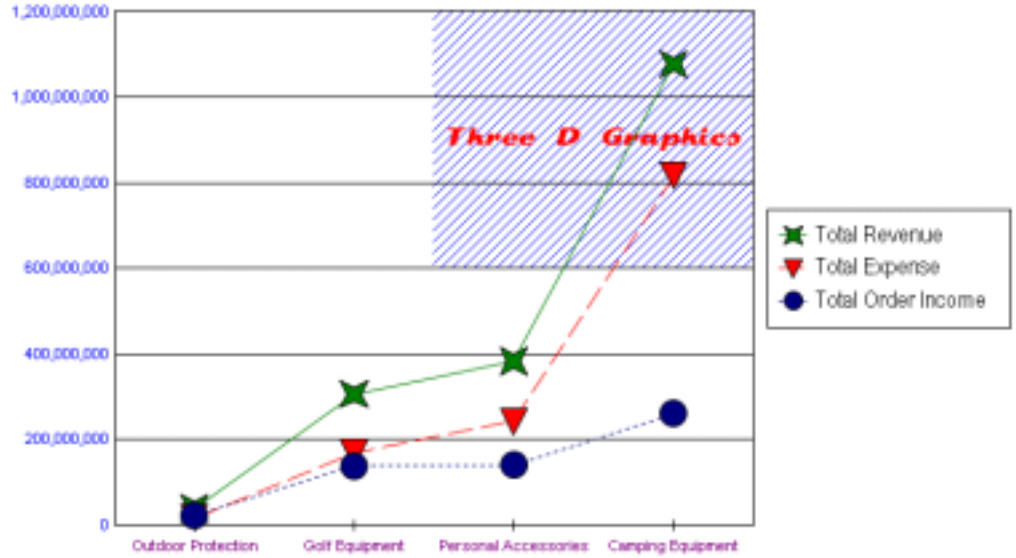
nPattern; -6...6. Positive values show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



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

EXAMPLE:

```
@USER_LABEL_FONT 14 255 0 0 255 Splash
@USER_FILL2 0.5 1.0 0.5 1.0 0 0 255 4 Three D Graphics
```



PERSISTENT:

NO

NOTES:

Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*). Charts Unlimited 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/*fStopX*; 0.0...1.1 X-Axis start/stop location

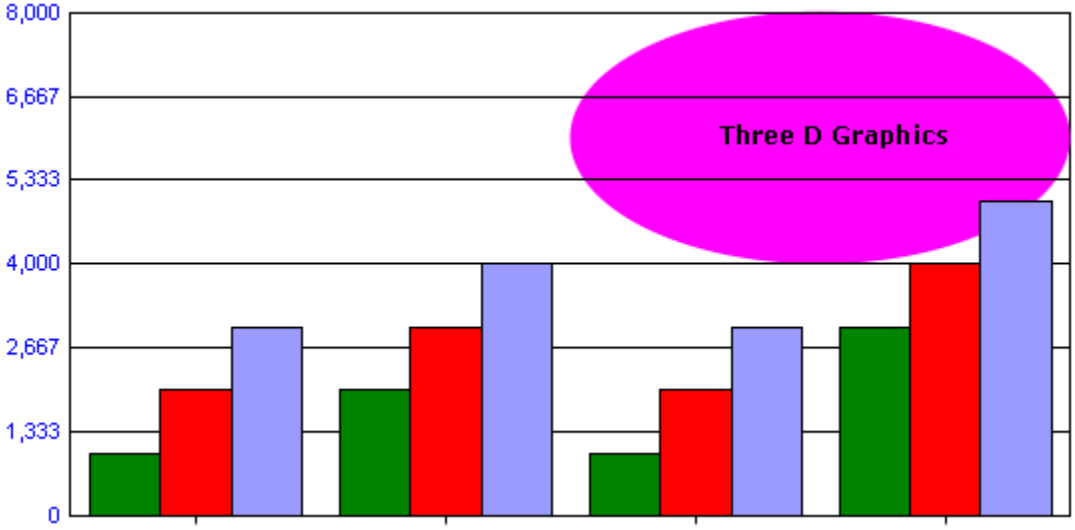
fStartY/*fStopY*; 0.0...1.1 Y-Axis start/stop location

nRed/*nGreen*/*nBlue*; 0...255: Define the circle fill color.

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

EXAMPLE:

```
@USER_FILL_CIRCLE 0.5 1.0 0.5 1.0 255 0 255 Three D Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).
- Charts Unlimited 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. The circle is drawn above the plot area. 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 location

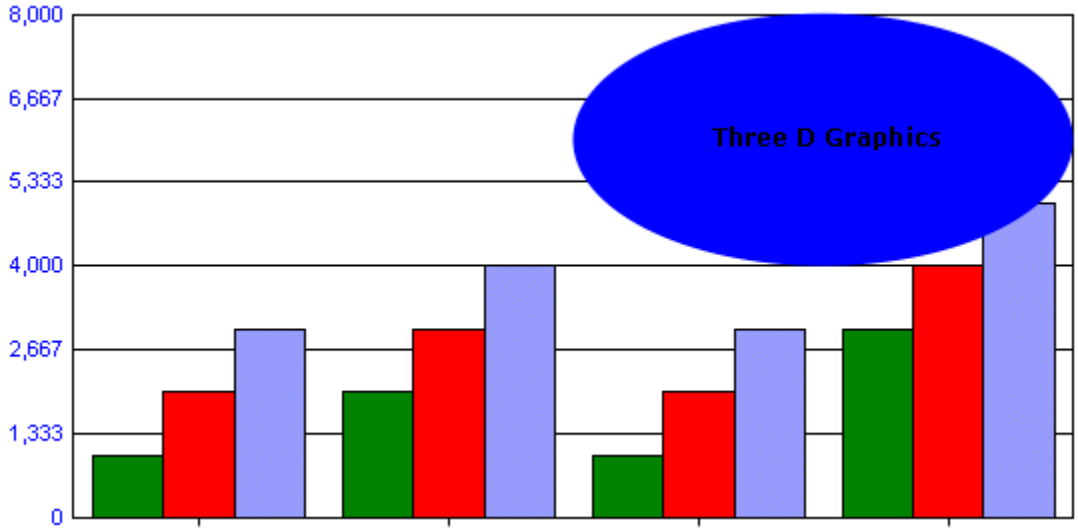
fStartY/*fStopY*; 0.0...1.1 Y-Axis start/stop location

nRed/*nGreen*/*nBlue*; 0...255: Define the circle fill color

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

EXAMPLE:

```
@USER_FILL_CIRCLE_ABOVE 0.5 1.0 0.5 1.0 0 0 255 Three D Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).
- Charts Unlimited 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
```

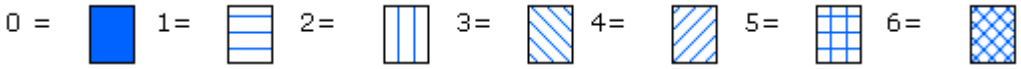
PARAMETERS:

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

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

nRed/*nGreen*/*nBlue*; 0...255: Defines the pattern color.

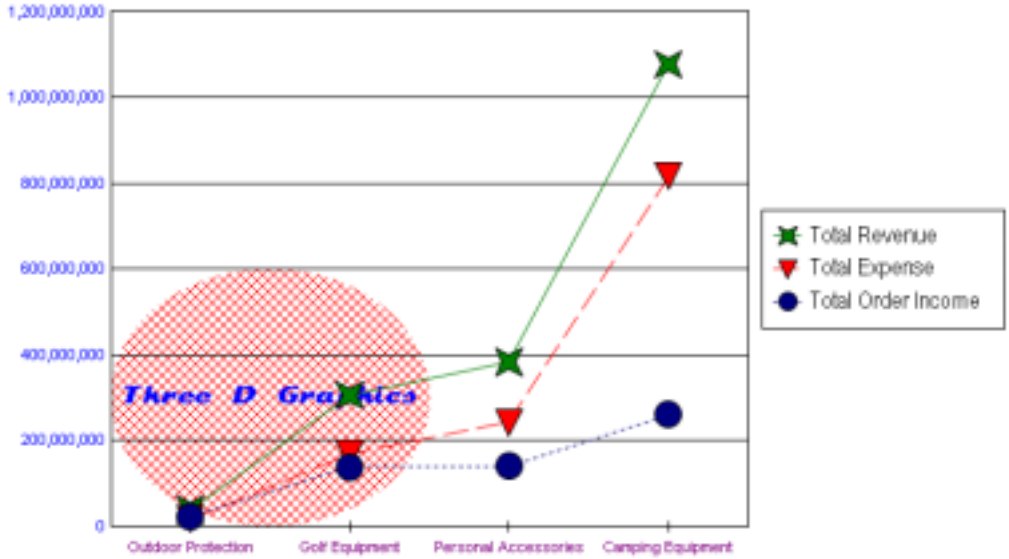
nPattern; -6...6. Positive values show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



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

EXAMPLE:

```
@USER_LABEL_FONT 14 0 0 255 255 Splash
@USER_FILL_CIRCLE2 0.0 0.5 0.0 0.5 255 0 0 6 Three D Graphics
```



PERSISTENT:

NO

NOTES:

Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*). Charts Unlimited 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. The circle is drawn above the plot area. 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
```

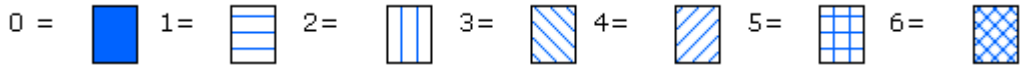
PARAMETERS:

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

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

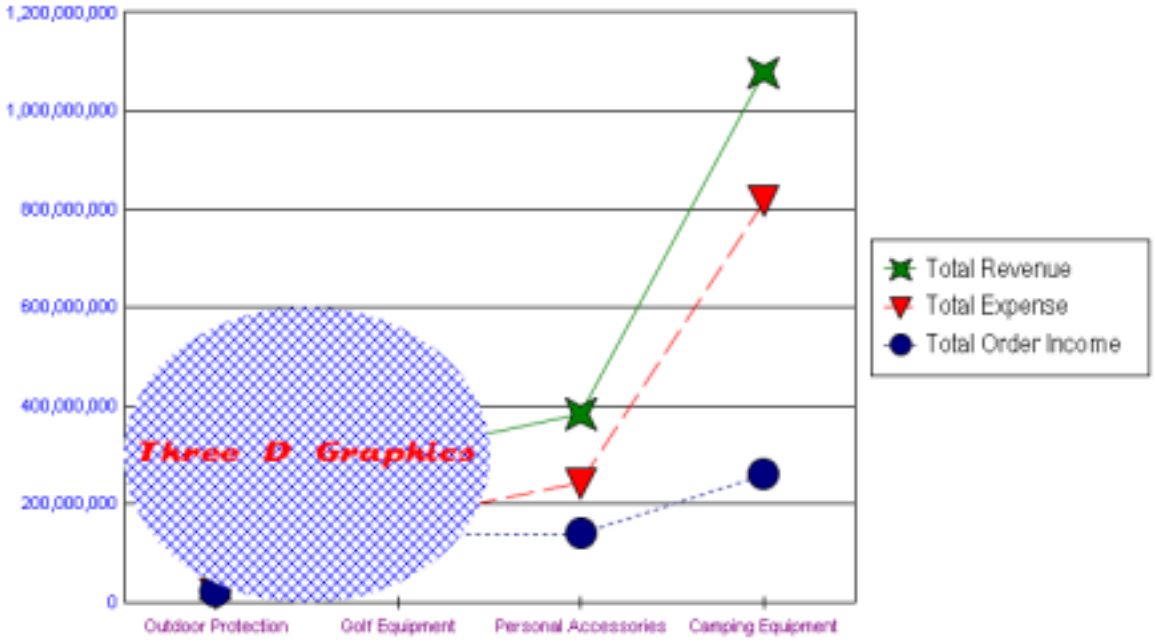
nRed/*nGreen*/*nBlue*; 0...255: Defines the pattern color.

nPattern; -6...6. Positive values 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 to draw in the middle of the circle. Add a tilde character (~) to this string if you intend to define another macro in the same title field.

```
@USER_LABEL_FONT 14 255 0 0 255 Splash
@USER_FILL_CIRCLE2_ABOVE 0.0 0.5 0.0 0.5 0 0 255 6 Three D
Graphics
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).

- Charts Unlimited 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.

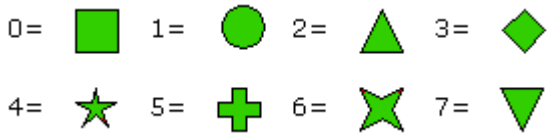
SYNTAX:

```
@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:

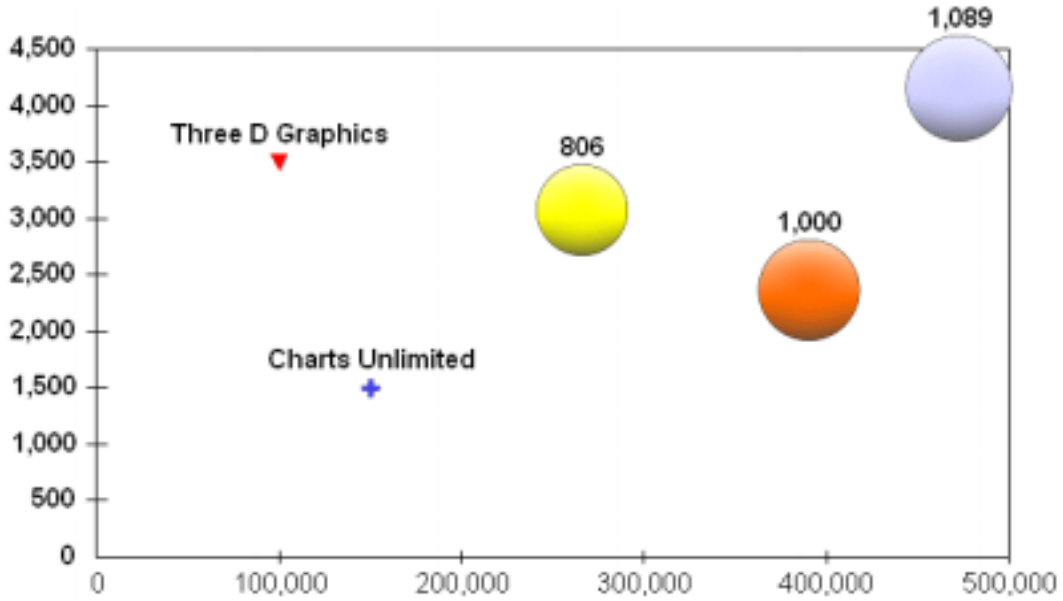


nRed; 0...255, *nGreen;* 0...255, *nBlue;* 0...255 defines the color of the marker.

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

EXAMPLE:

```
@USER_MARKER 100000 3500 7 255 0 0 Three D Graphics
@USER_MARKER 150000 1500 5 0 0 255 Charts Unlimited
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of eight user markers defined with @USER_MARKER and @USER_MARKER2.

ALSO SEE:

@USER_MARKER2

@USER_MARKER2 (User-Defined Marker with Value)

This macro adds a user-defined marker with optional text to a chart. With this macro, you can also specify a null marker and a value to append to the phrase (*szPhrase*). *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.

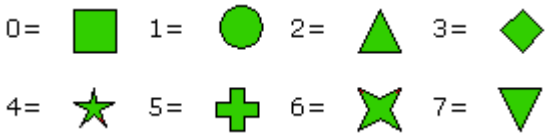
SYNTAX:

```
@USER_MARKER2 fX fY nShape nRed nGreen nBlue fValue szPhrase
```

PARAMETERS:

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

nShape; 0...8. 0...7 selects one of the following markers:



8=null/no marker

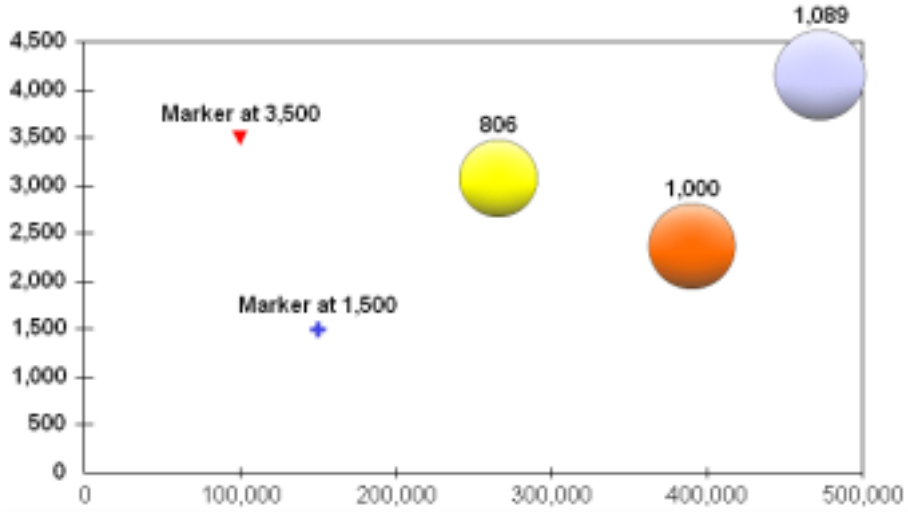
nRed; 0...255, *nGreen;* 0...255, *nBlue;* 0...255 defines the color of the marker.

fValue; Value to append to *szPhrase*.

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

EXAMPLE:

```
@USER_MARKER2 100000 3500 7 255 0 0 3500 Marker at ~
@USER_MARKER2 150000 1500 5 0 0 255 1500 Marker at
```



PERSISTENT:

NO

NOTES:

Charts Unlimited supports a maximum of eight user markers defined with @USER_MARKER and @USER_MARKER2.

ALSO SEE:

@USER_MARKER

@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/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 defines the rectangle outline color

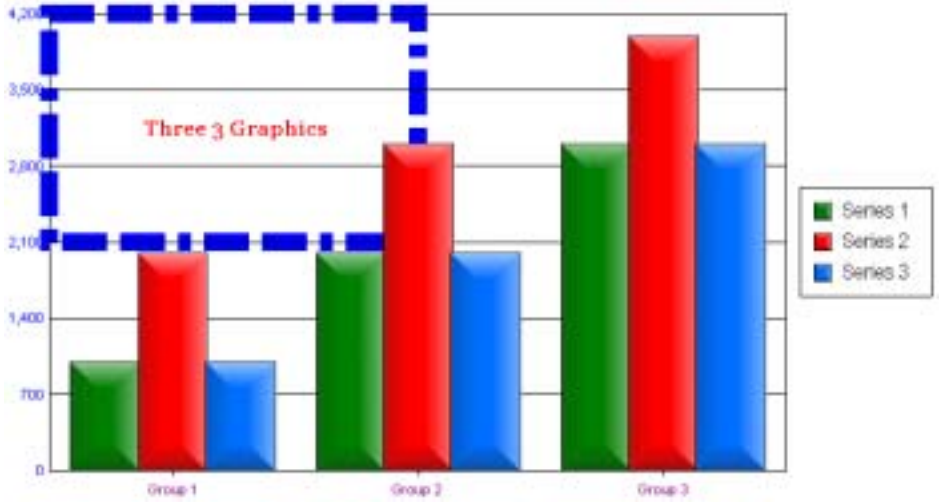
nLineStyle; 0...15 selects the line style (0=Solid, 1=Dashed, 2=Dotted, 3=Dot-Dash, 4=Dash-Dot-Dot, 5=Medium Dash, 6=Short Dash, 7=Long Dash, 8=Long Dot, 9=Dot-Dot-Dot, 10=Dash-Dash-Dot, 11=Dash-Dash-Dot-Dot, 12=Long Dash-Dot, 13=Long Dash-Dot-Dot, 14=Long Dash-Dash-Dot, 15=Long Dash-Dash-Dot-Dot)

nLineThickness; 0...1000 Line Thickness

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

EXAMPLE:

```
@USER_LABEL_FONT 12 255 0 0 255 Georgia
@USER_RECT 0.0 0.5 1.0 0.5 0 0 255 14 1000 Three D Graphics
```



PERSISTENT:

NO

NOTES:

Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*). Charts Unlimited 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:

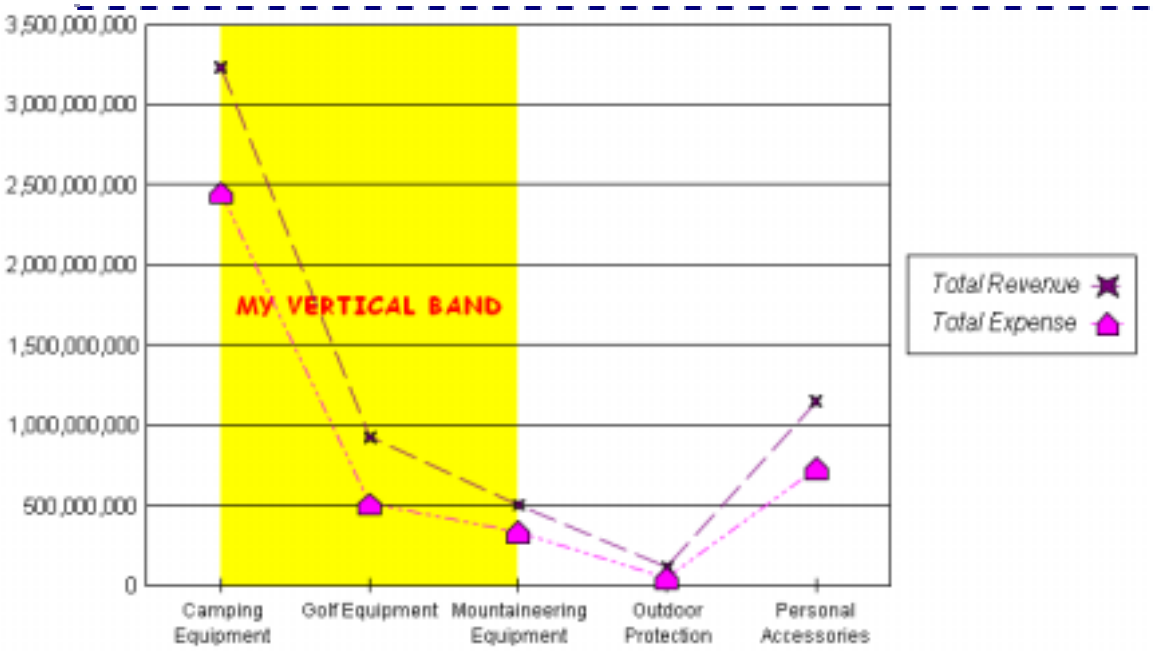
fStartX; 0.0...1.0 X-Axis start location

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 .1 .5 MY VERTICAL BAND
@WC 255 255 0
@USER_LABEL_FONT 12 255 0 0 255 Comic Sans MS
```



PERSISTENT:

NO

NOTES:

- Use the @USER_LABEL_FONT macro to format the optional label (*szPhrase*).
- Charts Unlimited 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:

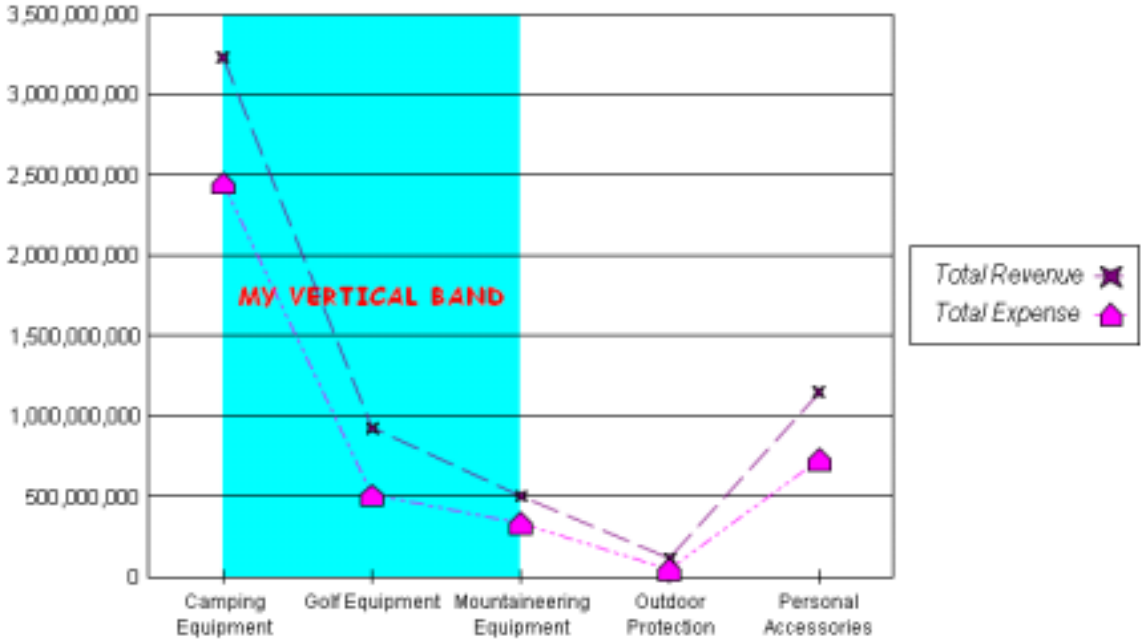
```
@WC nRed nGreen nBlue
```

PARAMETERS:

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

SYNTAX:

```
@UW .1 .5 MY VERTICAL BAND
@WC 0 255 255
@USER_LABEL_FONT 12 255 0 0 255 Comic Sans MS
```



PERSISTENT:

NO

ALSO SEE:

@UW


THREE D GRAPHICS


CHARTS
UNLIMITED

Section 11: Chart Type Macros

These macros change the chart type:

- @3DSCAT; Change the chart type to a 3D Scatter Chart
- @COMBO; Change the chart type to a Combo Chart
- @COMPARE2; Change the chart type to a 2-Series Absolute Bar Chart
- @GANTT; Change the chart type to a Gantt Chart
- @GRAPHTYPE; Select a Graph Type
- @PARETO; Change the chart type to a Pareto Chart
- @POLAR; Change the chart type to a Polar Chart
- @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:

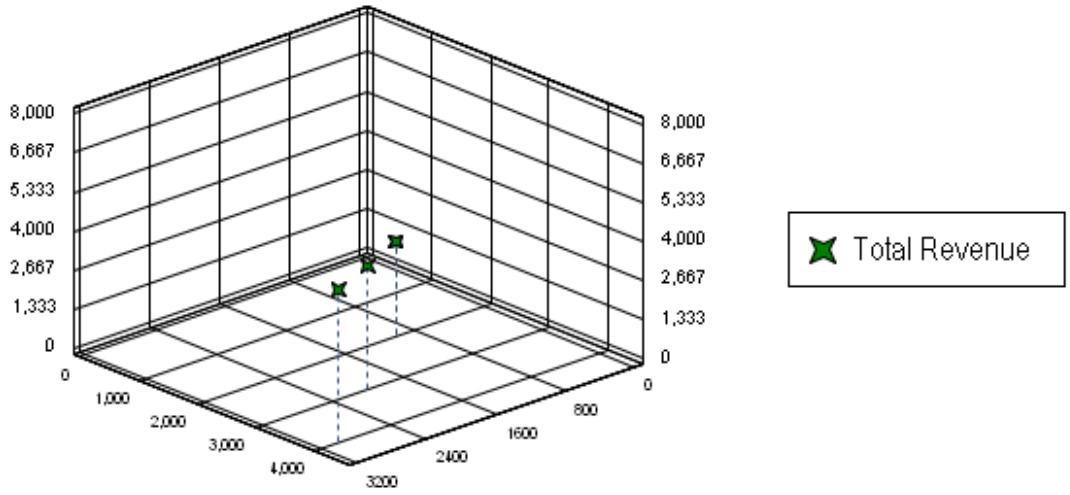
```
@3DSCAT
```

PARAMETERS:

None

EXAMPLE:

```
@3DSCAT
```



PERSISTENT:

YES

@COMBO (Combo Chart)

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

SYNTAX:

```
@COMBO nSeries nStyle
```

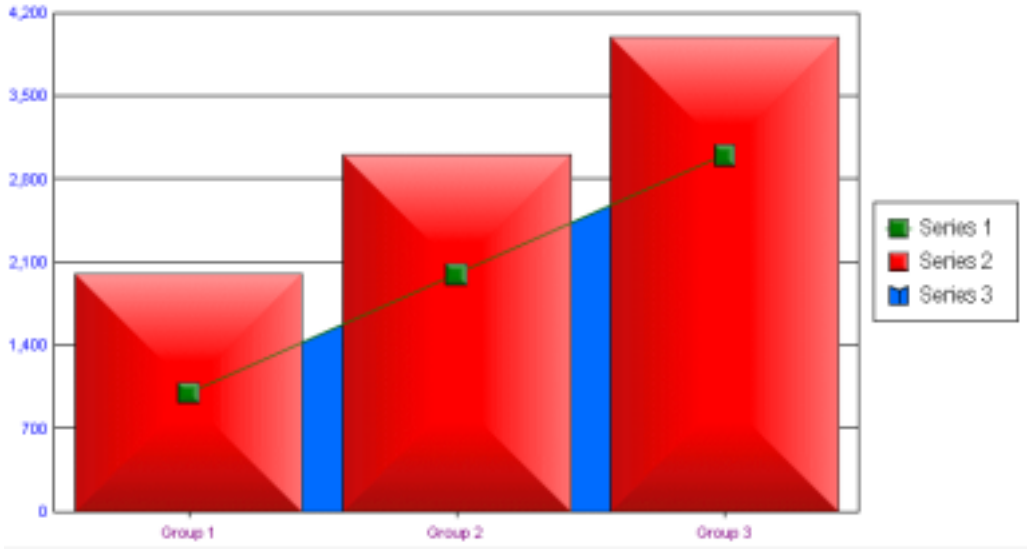
PARAMETERS:

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.

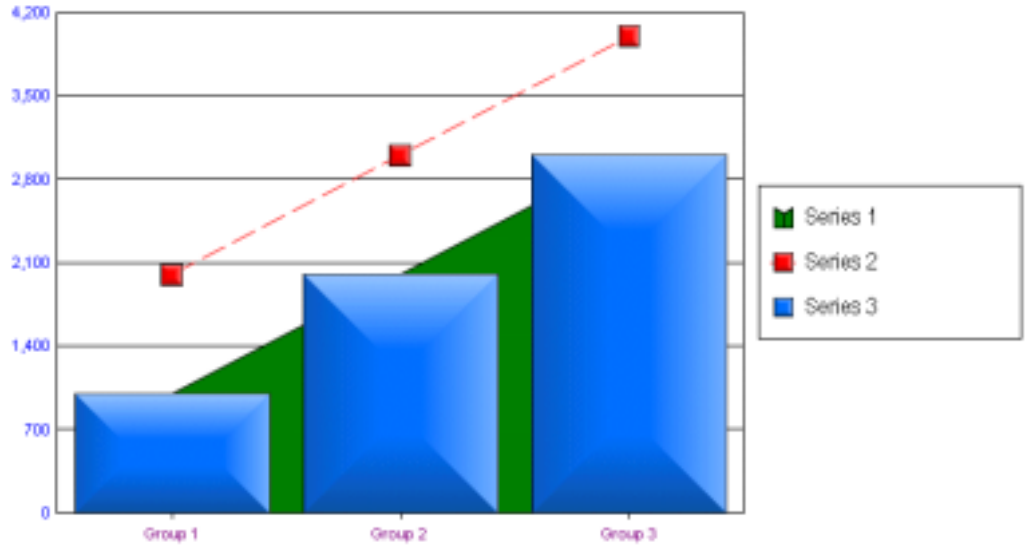
nStyle; 0...2 (0= Bar, 1=Line, 2=Area)

EXAMPLE:

```
@COMBO 0 1 @COMBO 1 0 @COMBO 2 2 @BEVEL 3 1
```




```
@COMBO 0 2 @COMBO 1 1 @COMBO 2 0 @BEVEL 3 1
```



PERSISTENT:

YES

@COMPARE2 (2-Series Absolute Bar Chart)

This macro creates a 2-series absolute bar chart where the first series is about twice as wide as the second series.

SYNTAX:

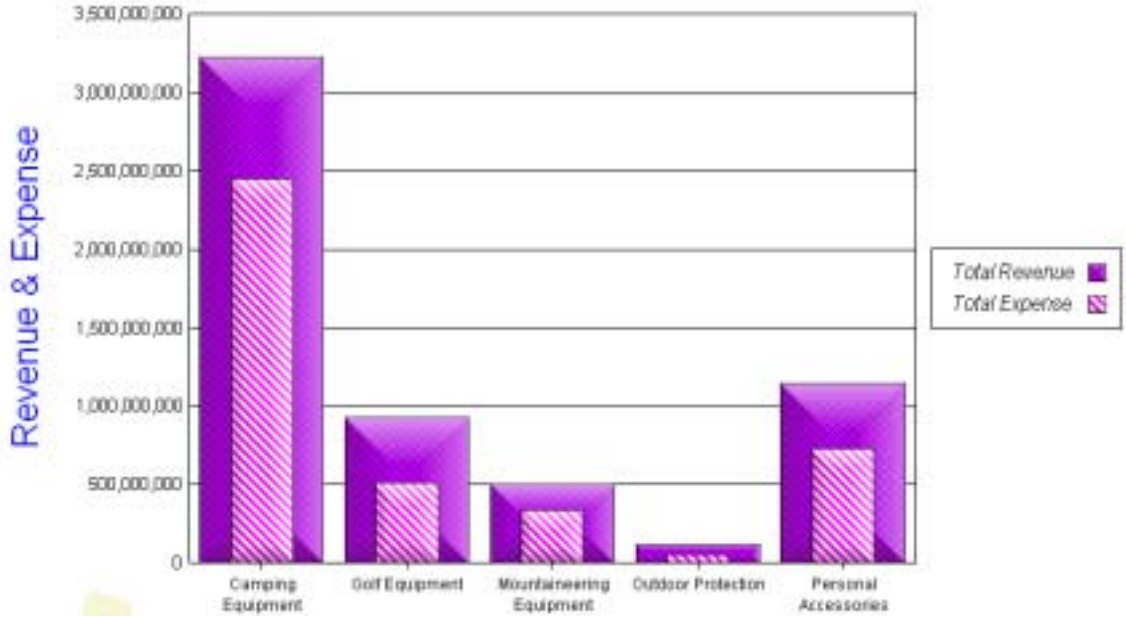
```
@COMPARE2 bCompare
```

PARAMETERS:

bCompare; 0/1

EXAMPLE:

```
@COMPARE2 1
```



PERSISTENT:

YES

@GANTT (Gantt Chart)

This macro creates a Gantt Chart with the specified parameters. A Gantt Chart is intended to show the status of tasks in a project between a scheduled start date and stop date or start date and duration.

SYNTAX:

```
@GANTT bTimeAxis bDurationMode bGroupCompress
```

PARAMETERS:

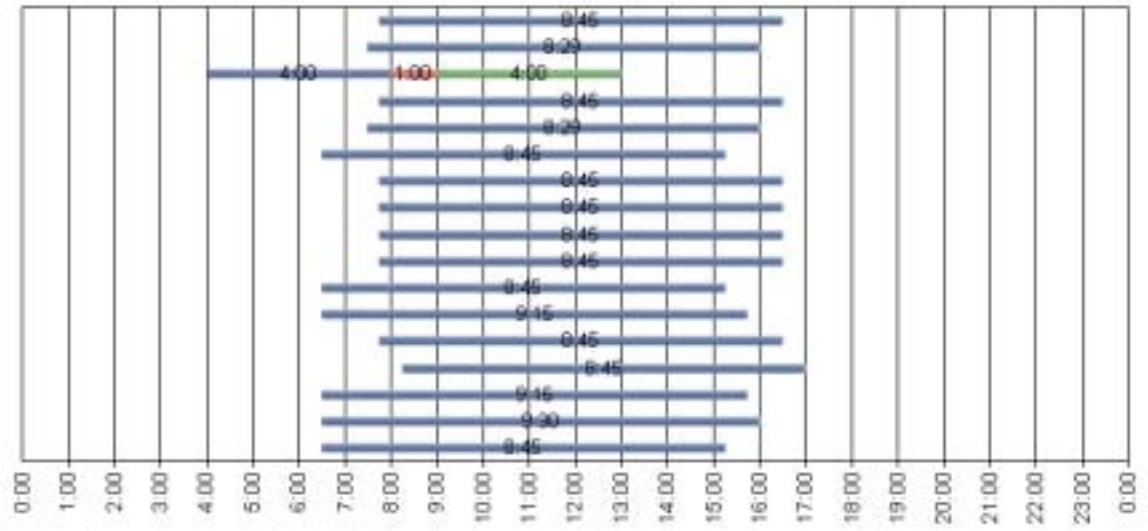
bTimeAxis; 1 = use time axis, 0 = do not use time axis

bDurationMode; 1 = Gantt data is Start, Duration, 0 = Gantt data is Start, Stop

bGroupCompress; 1 = use group labels to calculate the number of blocks per group entry, 0 = do not use group labels to calculate the number of blocks per group entry

EXAMPLE:

```
@GANTT 0 0 1
```



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 ***n***

PARAMETERS:

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

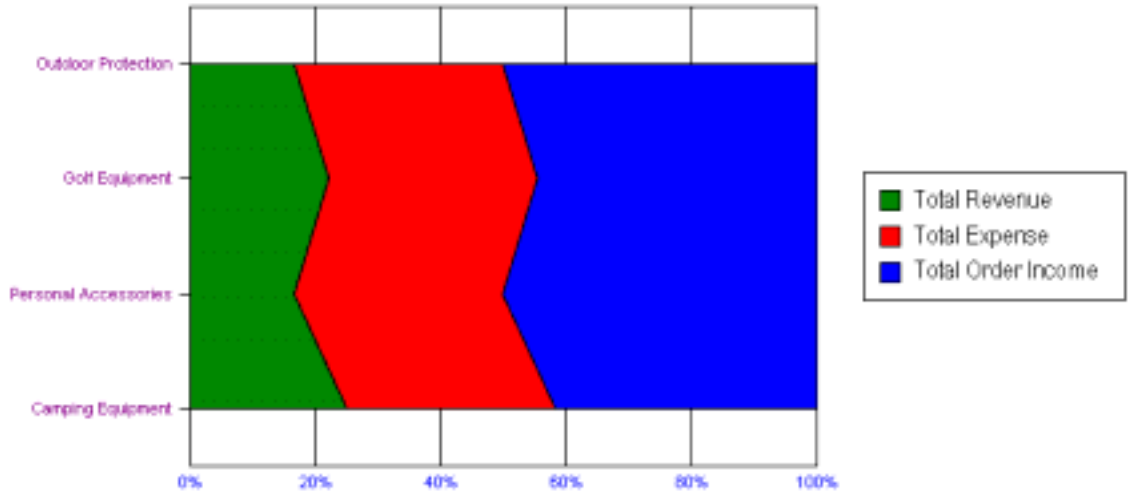
Value	Chart
0	Vertical Area Absolute
1	Vertical Area Stacked
2	Vertical Area Bi-Polar Absolute
3	Vertical Area Bi-Polar Stacked
4	Vertical Area Dual-Y Absolute
5	Vertical Area Dual-Y Stacked
6	Vertical Area Percent
7	Horizontal Area Absolute
8	Horizontal Area Stacked
9	Horizontal Area Bi-Polar Absolute
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

Value	Chart
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
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

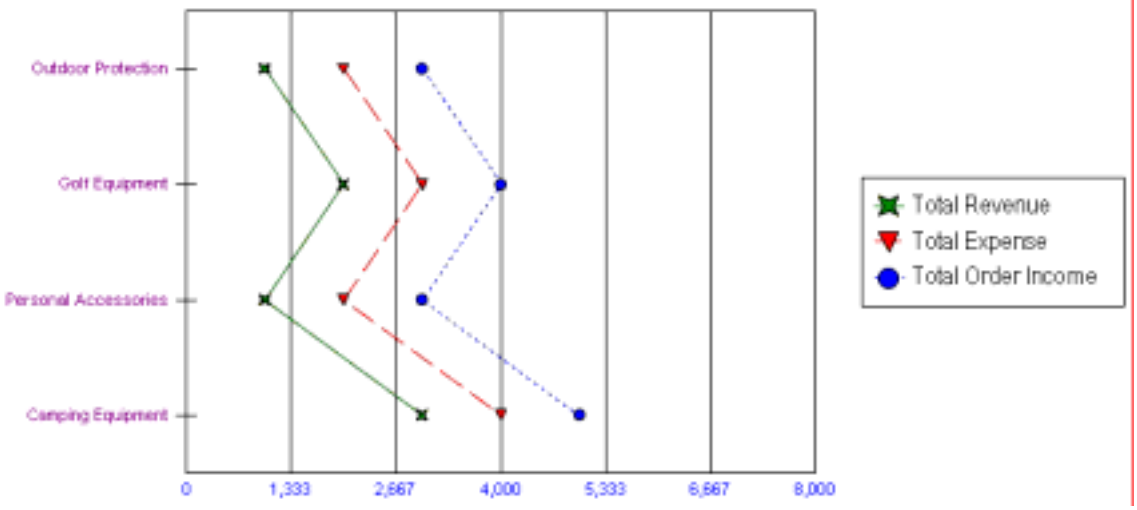
Value	Chart
64	High-Low Dual-Y Stock Chart
65	High-Low-Open Stock Chart
66	High-Low-Open Dual-Y Stock Chart
67	High-Low-Open-Close Stock Chart
68	High-Low-Open-Close Dual-Y Stock Chart
69	Spectral Map
70	Vertical Histogram
71	Horizontal Histogram
72	Table
73	3D Riser – Bars
74	3D Riser – Pyramids
75	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 13
```



@GRAPHTYPE 35



PERSISTENT:

YES

@PARETO (Pareto Chart)

This macro creates a Pareto chart. If *n* is zero, a simple Pareto chart is created. If *n* is one, a 'classic pareto chart' with a cumulative percentage line is created.

SYNTAX:

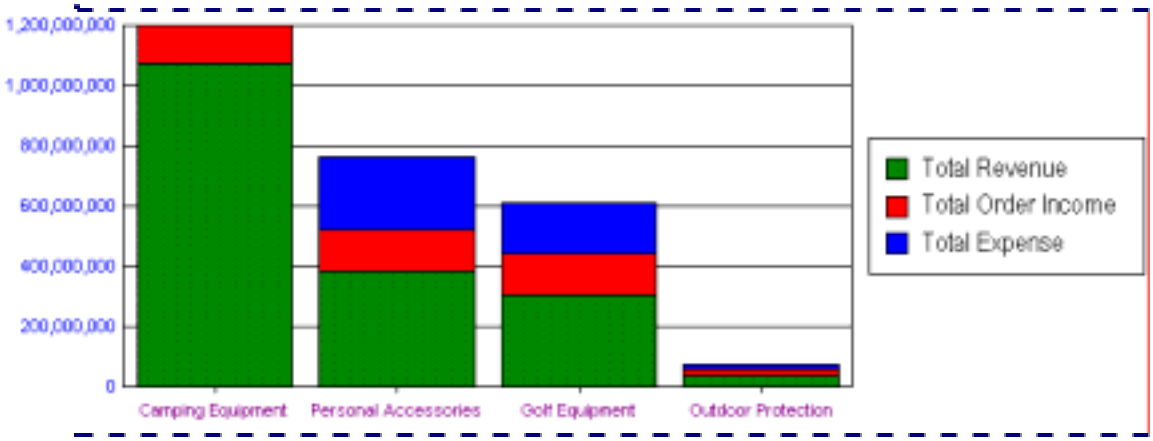
```
@PARETO n
```

PARAMETERS:

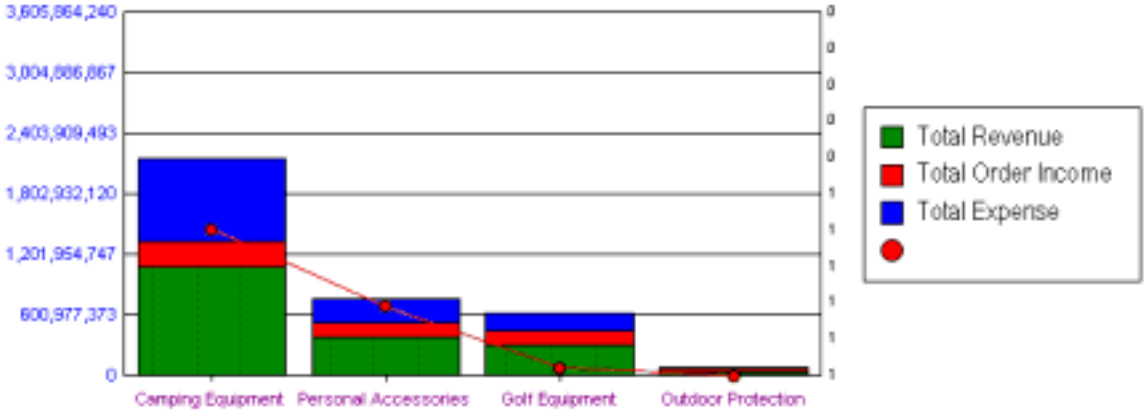
n; 0 = create a simple Pareto chart. 1 = create a “classic” Pareto chart with a cumulative percentage line.

EXAMPLE:

```
@PARETO 0
```




```
@PARETO 1
```



PERSISTENT:

YES

@POLAR (Polar Chart)

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

SYNTAX:

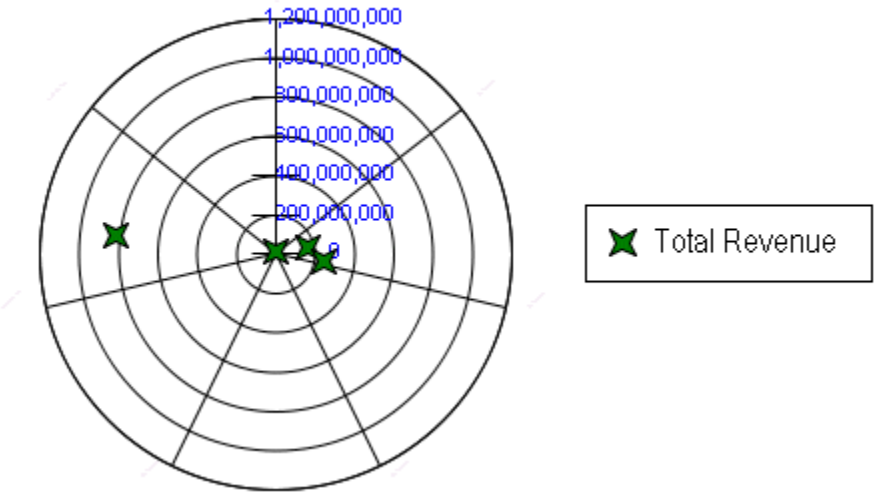
```
@POLAR
```

PARAMETERS:

None

EXAMPLE:

```
@POLAR
```



PERSISTENT:

YES

@RIVER (Floating Area Series)

In a three-series graph (normally line chart), this macro changes the first two series into a floating area (i.e., river) behind the third series. Set *bRiverMode*=1 to activate. Set the *bShowLegend* parameter to one to add the river area as an entry in the legend with its own *szRiverLabel* string.

SYNTAX:

```
@RIVER bRiverMode bShowLegend szRiverLabel
```

PARAMETERS:

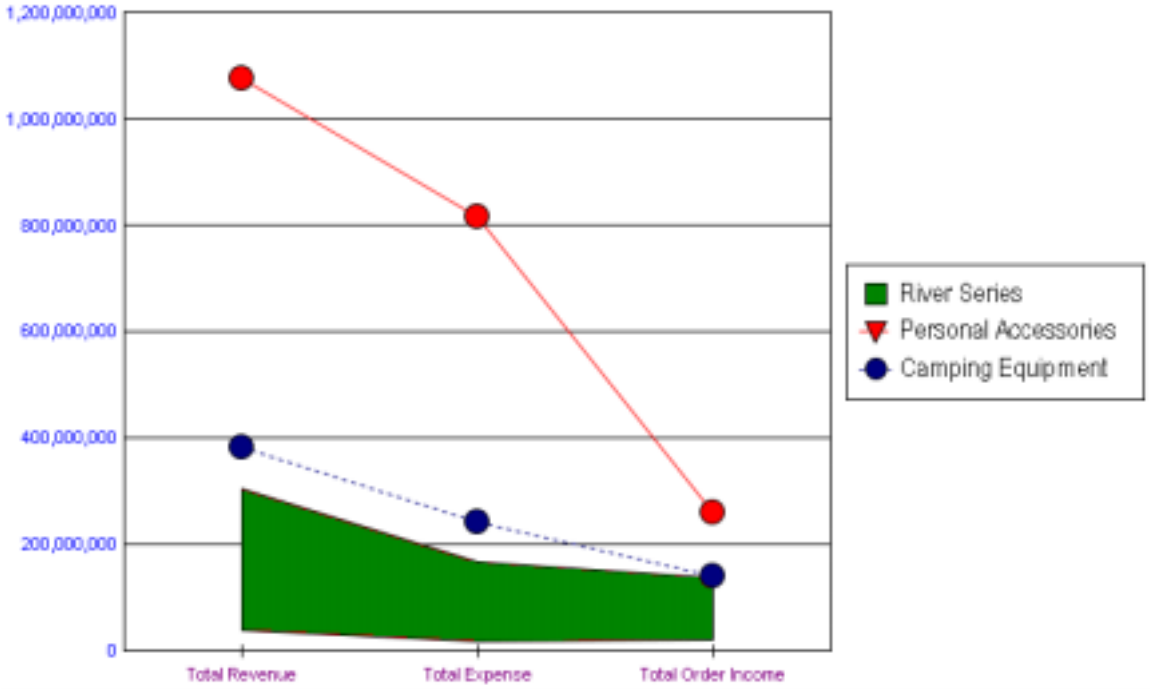
bRiverMode; 1=enable river mode, 0=disable river mode

bShowLegend; 1=add river area as an entry in the legend, 0=don't show river area in legend

szRiverLabel; label string to use in legend if *bShowLegend*=1

EXAMPLE:

```
@RIVER 1 1 River Series
```



PERSISTENT:

NO

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
- @IN; Move the First Box Plot In/Right by a specified number of virtual units
- @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:

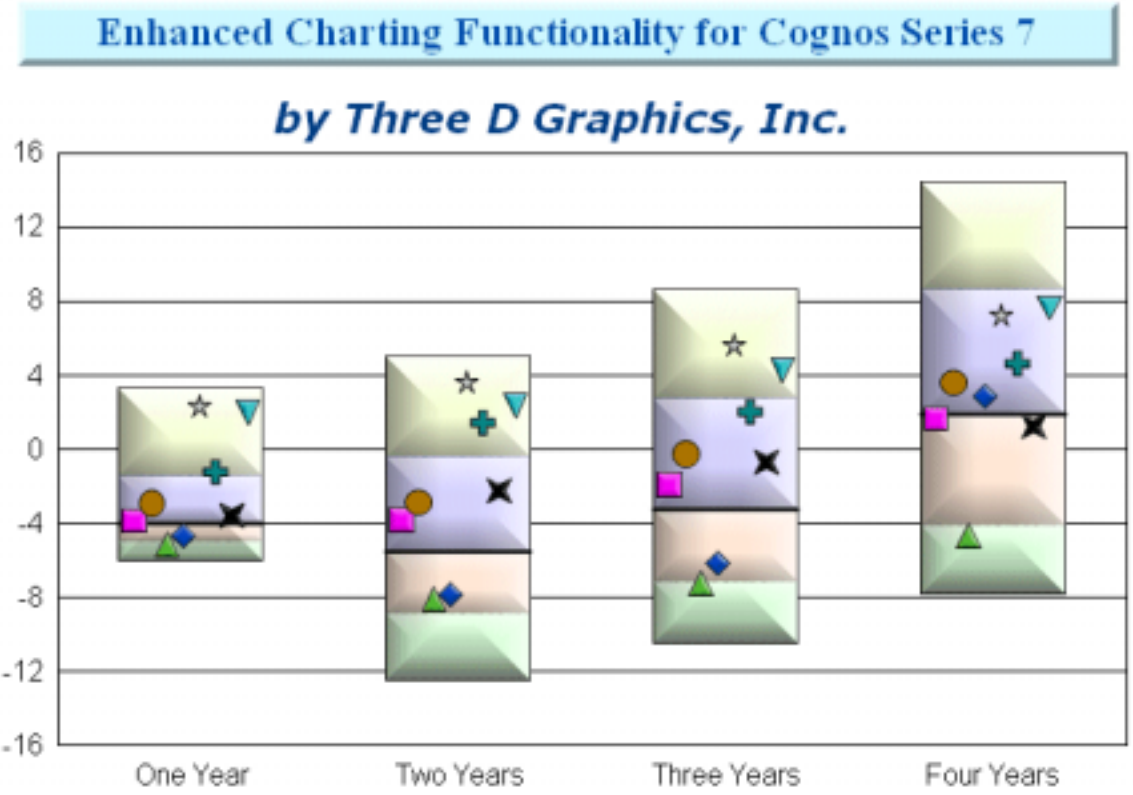
```
@BP or @BP1
```

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 with 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:

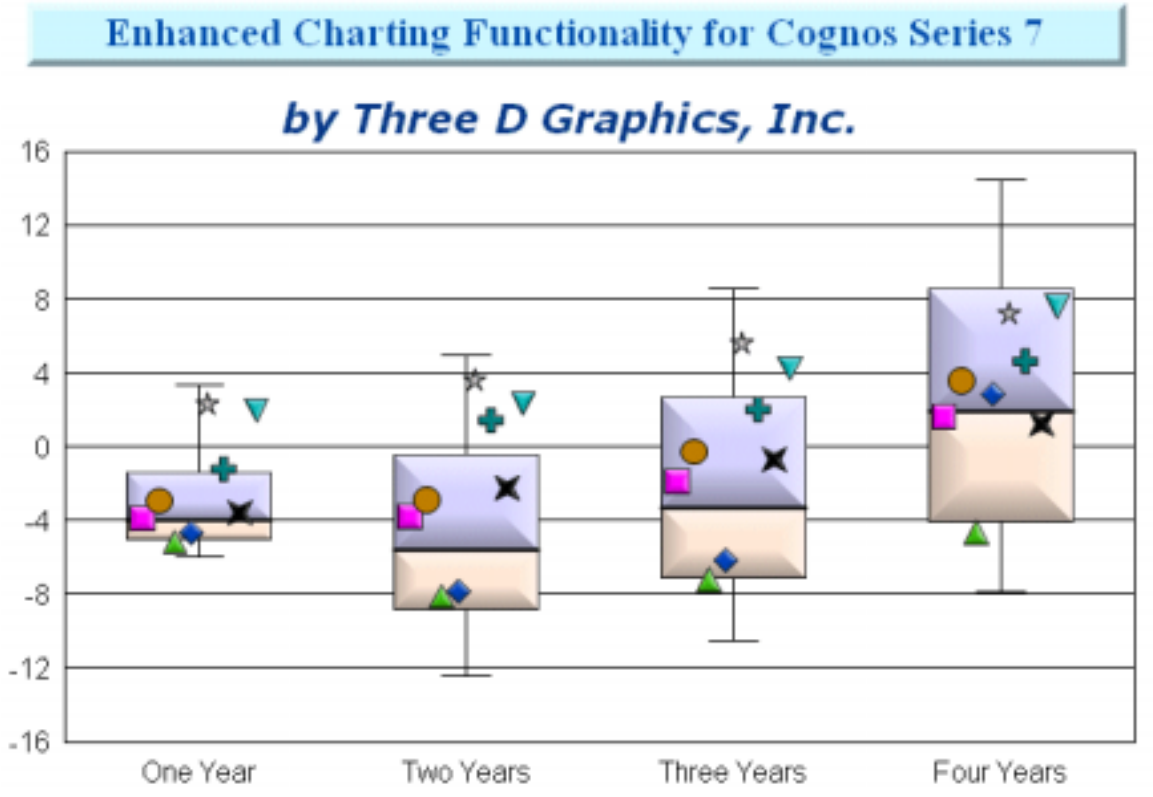
```
@BP2
```

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 with 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:

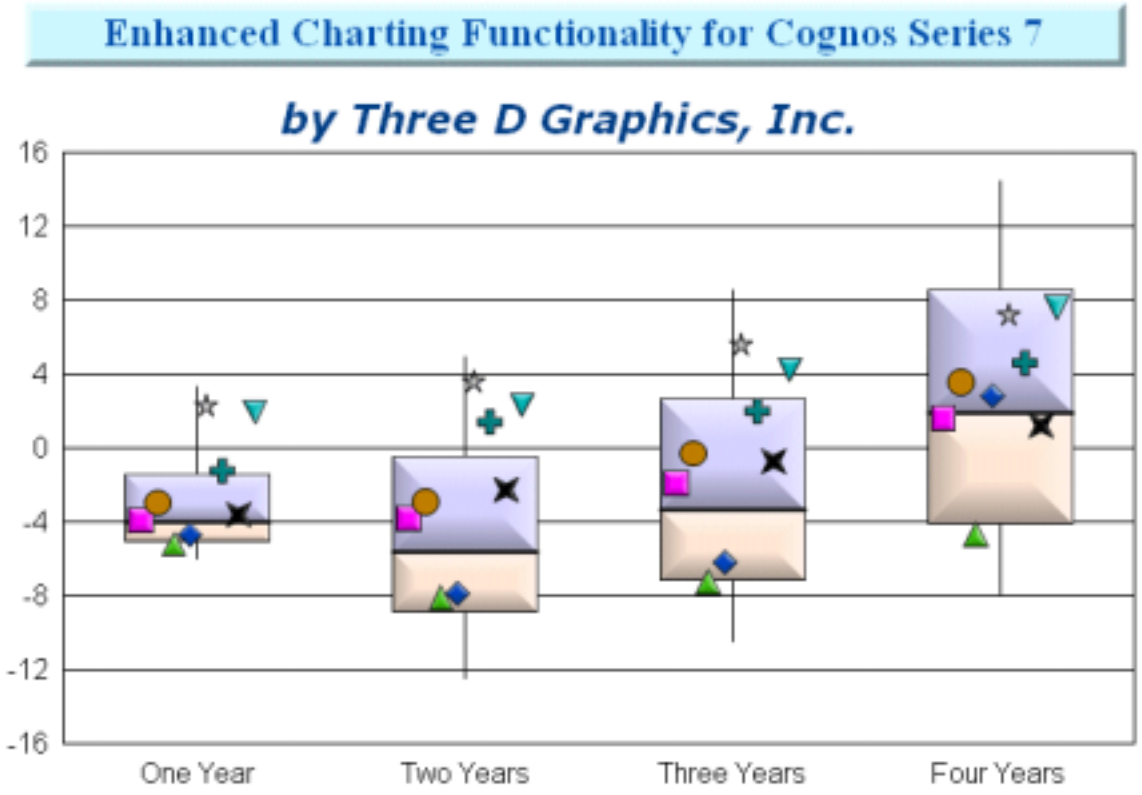
```
@BP3
```

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 with the @MS macro.

@BPH (Horizontal Box Plot)

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

SYNTAX:

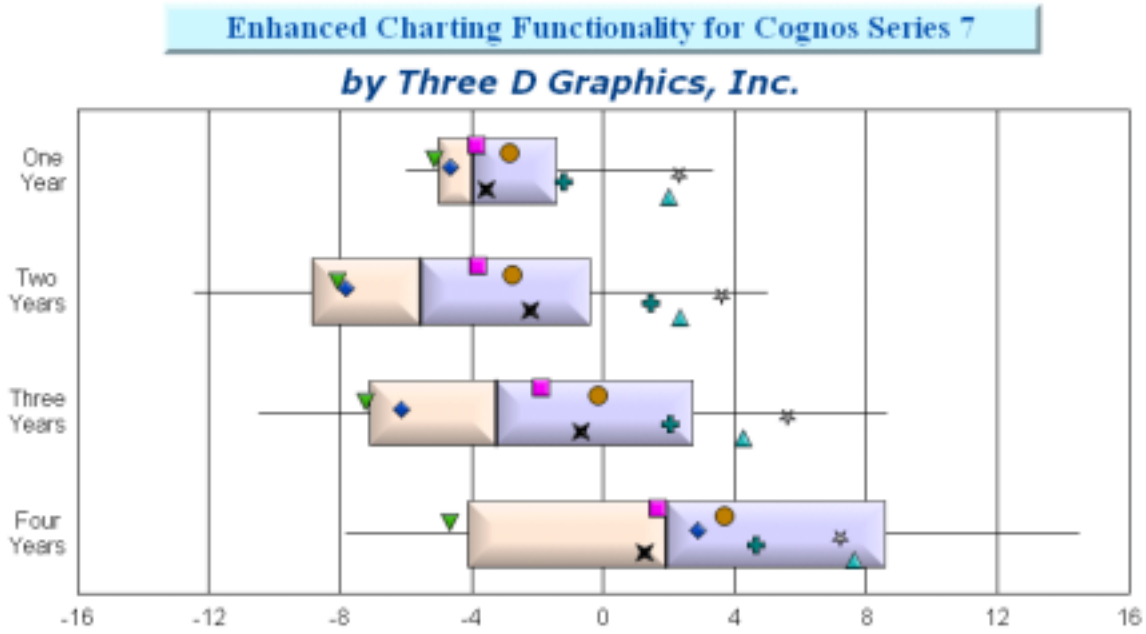
```
@BPH nOrient
```

PARAMETERS:

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

EXAMPLE:

```
@BP3
@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:

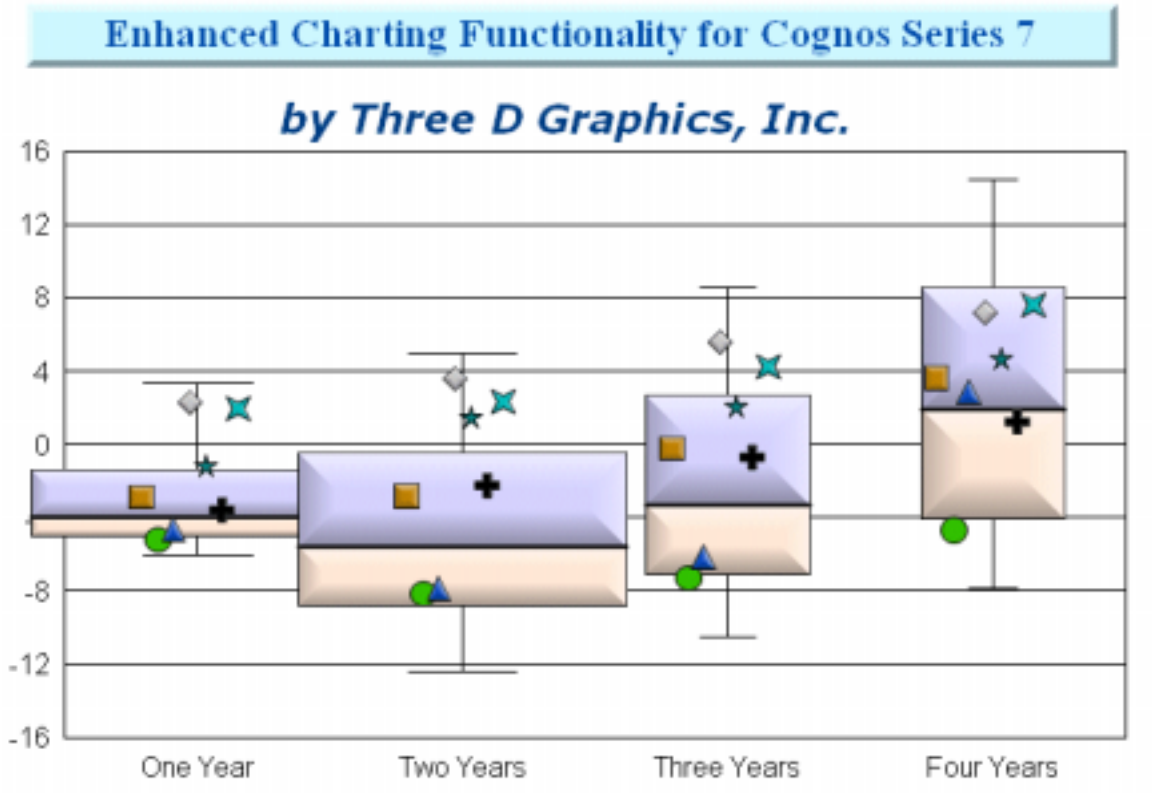
```
@BPW
```

PARAMETERS:

None

EXAMPLE:

```
@BPW
```



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP2, @BP3

@IN (Move First Box Plot In)

In a Box Plot Chart, this macro will move the first box to the right by a specified number of virtual units.

SYNTAX:

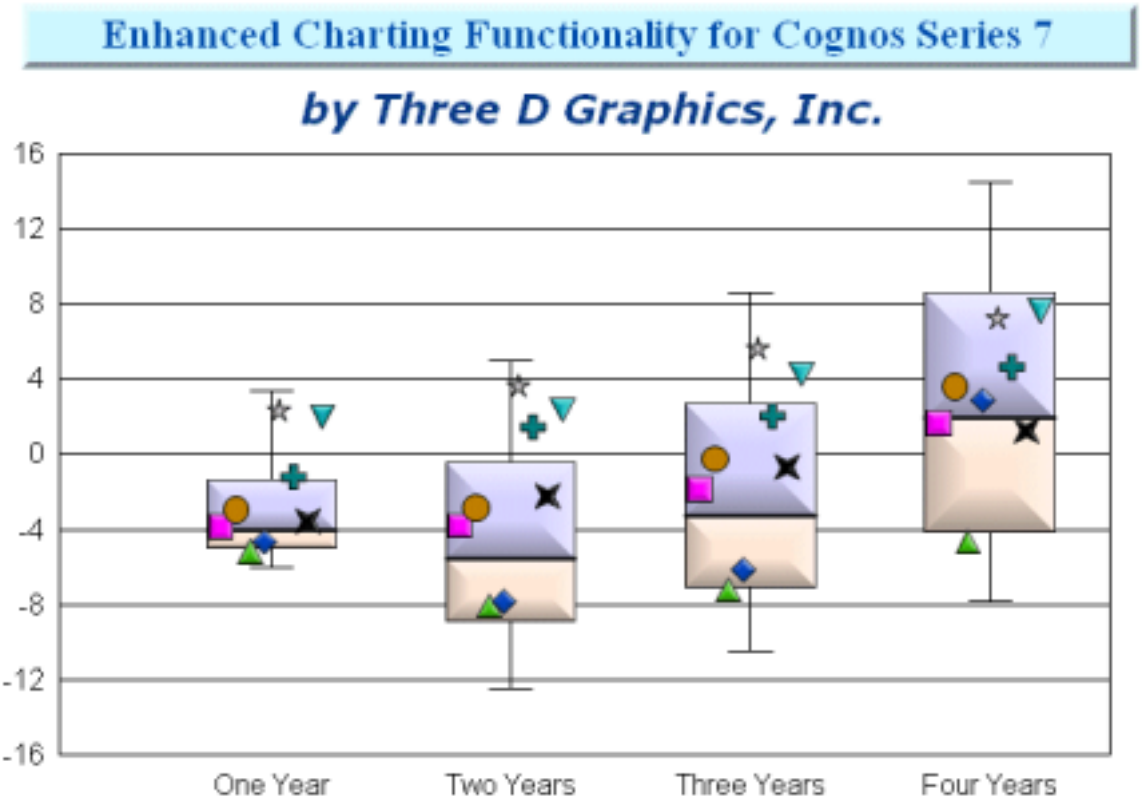
```
@IN n
```

PARAMETERS:

n; Number of virtual units (0...16000) to move the first box to the right.

EXAMPLE:

```
@BP2
@IN 2000
```



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP2, @BP3

@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)

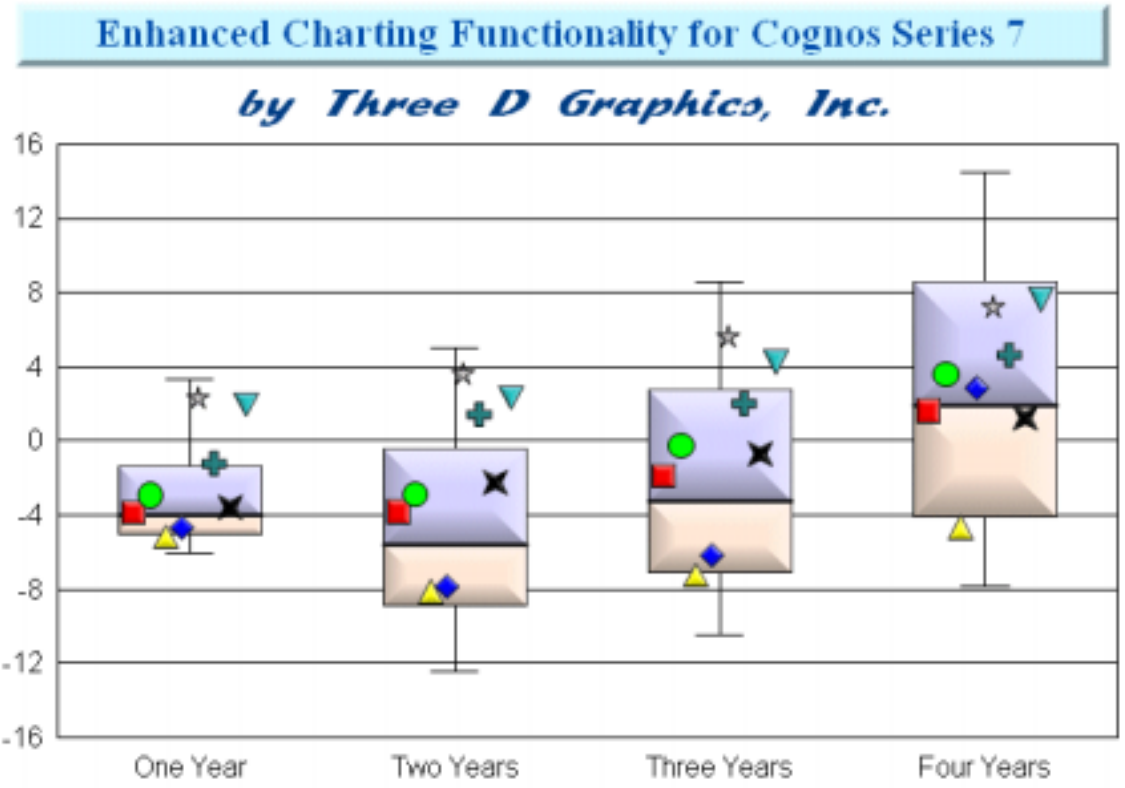
nRed; 0...255 defines the Red portion of RGB color selection.

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

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

EXAMPLE:

```
@BP2
@SZ 24
@MC 1 255 0 0
@MC 2 0 255 0
@MC 3 255 255 0
@MC 4 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:

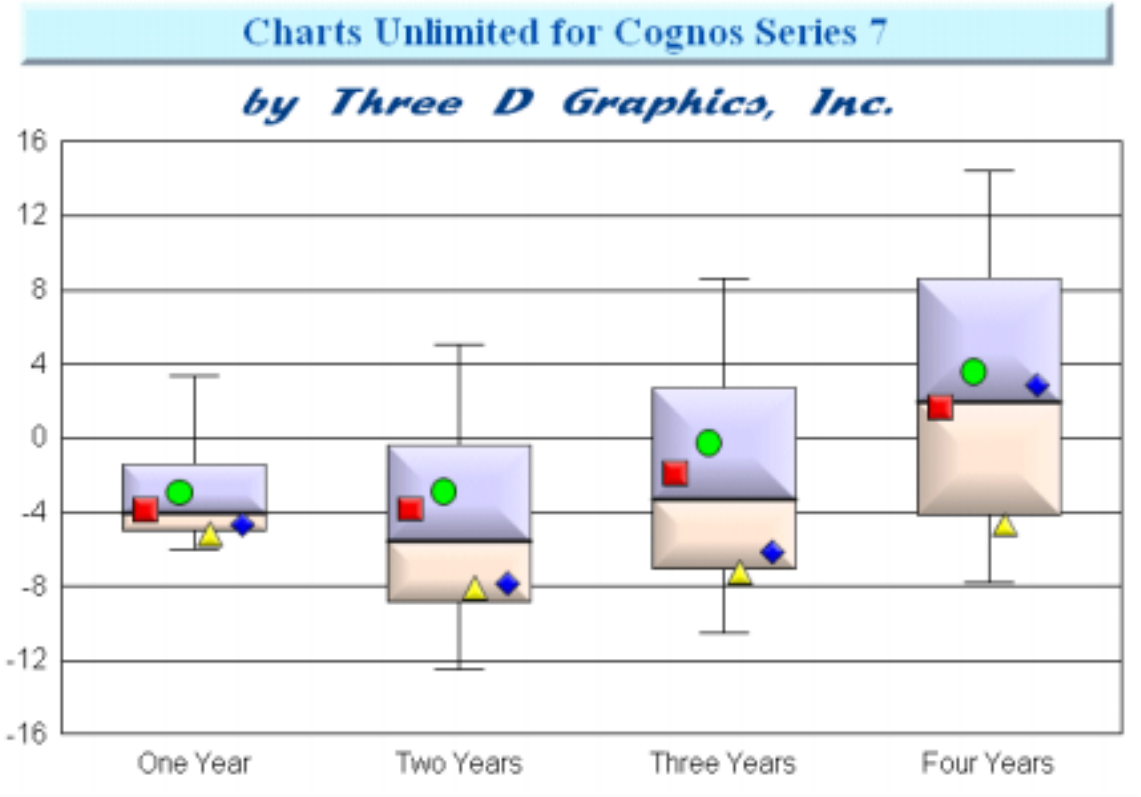
```
@MK nMarkers
```

PARAMETERS:

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

EXAMPLE:

```
@BP2
@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 with 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.

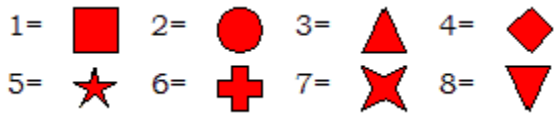
SYNTAX:

```
@MS nSeries nShape
```

PARAMETERS:

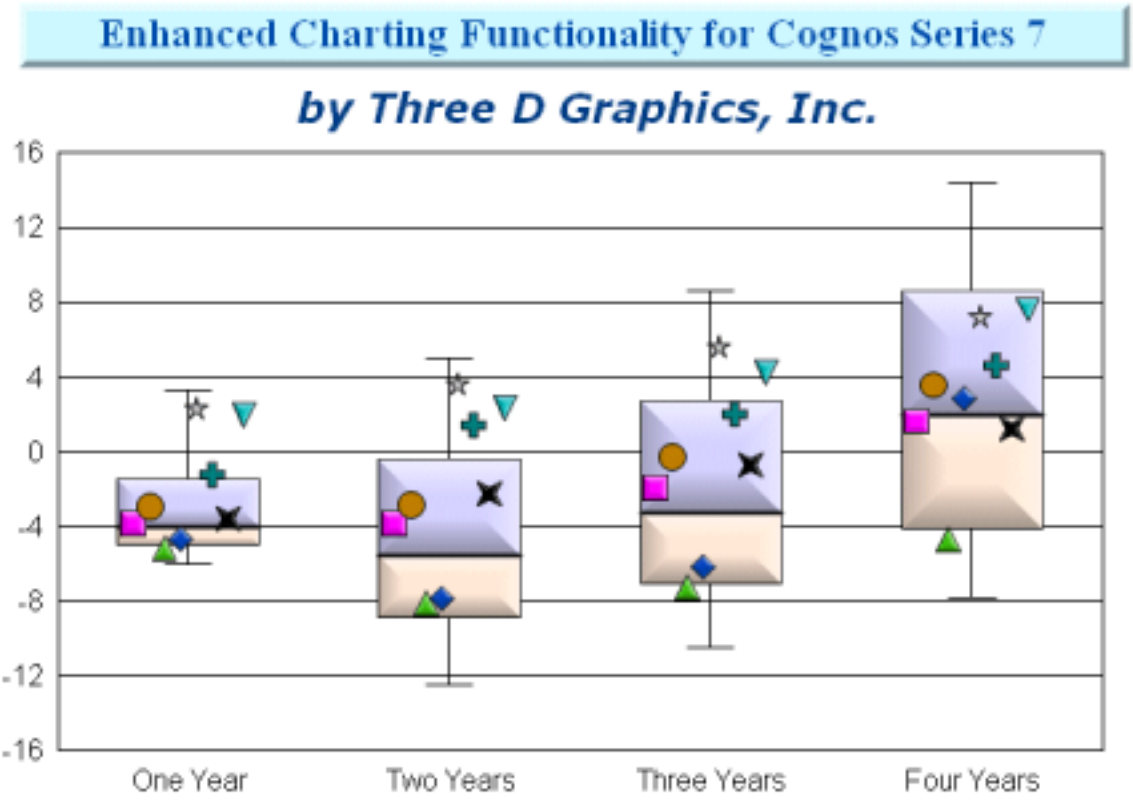
nSeries; Series Number (1...8).

nShape; 1...8 selects the shape to assign to series *nSeries*:



EXAMPLE:

```
@BP2 @SZ 24
@MS 1 8 @MS 2 7 @MS 3 6 @MS 4 5 @MS 5 4 @MS 6 3 @MS 7 2 @MS 8 1
```



PERSISTENT:

NO

NOTES:

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

Section 13: Macros for Gauges

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

- @GAUGE_ANGLE; Start/Stop Angle of Gauge labels/gridlines
- @GAUGE_BORDER_STYLE; Gauge border
- @GAUGE_BORDER_THICKNESS; Gauge border thickness
- @GAUGE_CHART; Create a Gauge Chart
- @GAUGE_MULTIPLE_NEEDLES; Gauge multiple needles
- @GAUGE_NEEDLE_STYLE; Gauge needle style
- @GAUGE_RANGE_COLOR; Gauge band colors
- @GAUGE_RANGE_START_STOP; Location on gauge axis where range starts/stops
- @GAUGE_RANGE_THICKNESS; Thickness of internal gauge bands
- @GAUGE_RANGE_THRESHOLD; Gauge Band Thresholds

@GAUGE_ANGLE (Gauge Angle)

This macro sets the start/stop angle for the labels and gridlines on the face of the gauge.

SYNTAX:

```
@GAUGE_ANGLE nStartAngle nStopAngle
```

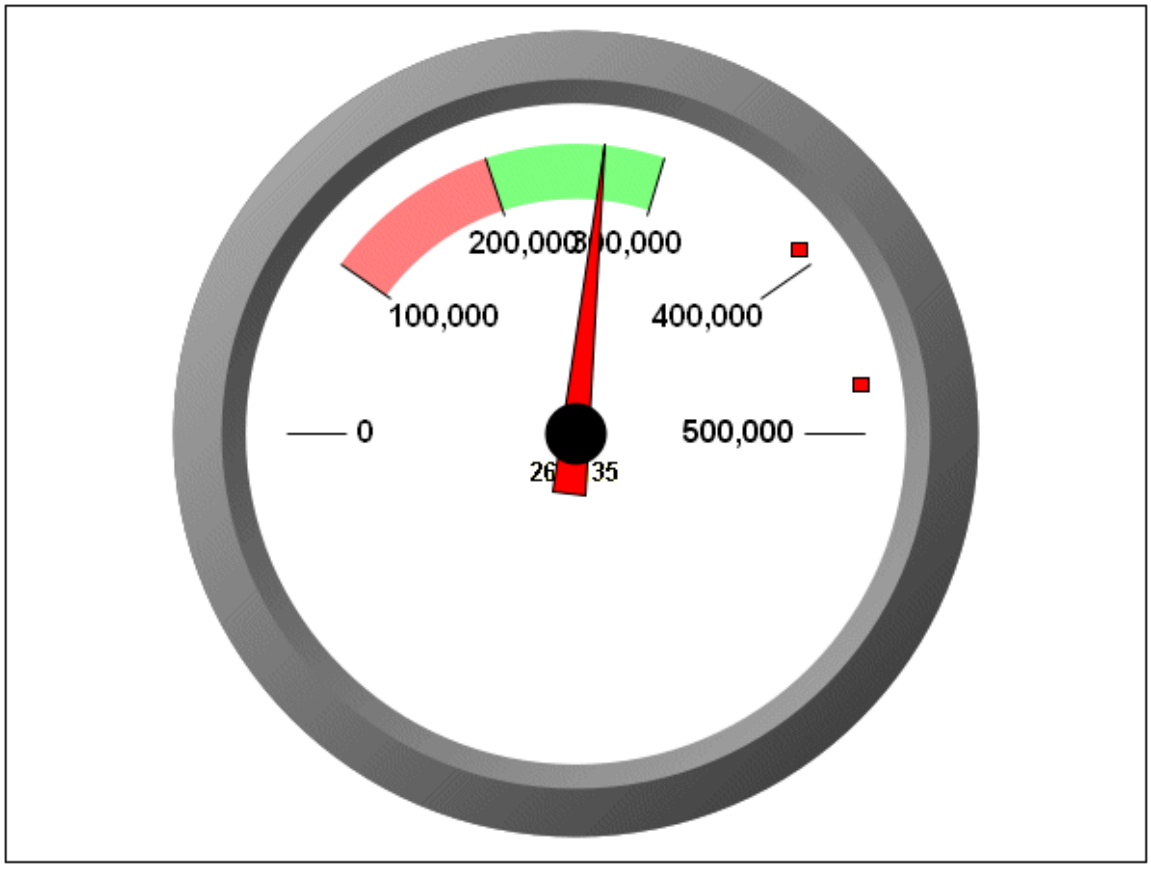
PARAMETERS:

nStartAngle; 0...360 Start Angle

nStopAngle; 0...360 Stop Angle

EXAMPLE:

```
@GAUGE_ANGLE 0 180
```



PERSISTENT:

NO

@GAUGE_BORDER_STYLE (Gauge Border Style)

This macro can be used to select a gauge border style.

SYNTAX:

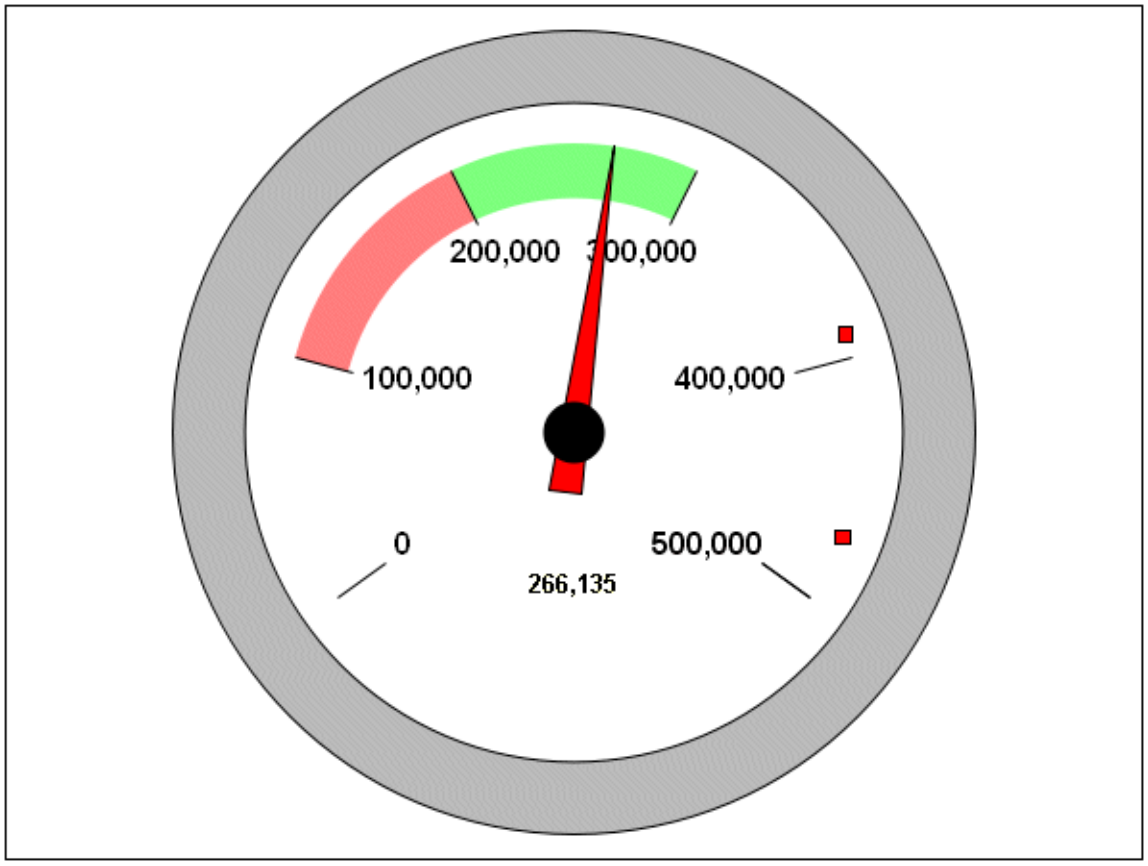
```
@GAUGE_BORDER_STYLE nStyle
```

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 1
```



PERSISTENT:

NO

@GAUGE_BORDER_THICKNESS (Gauge Border Thickness)

This macro controls the thickness of the gauge border.

SYNTAX:

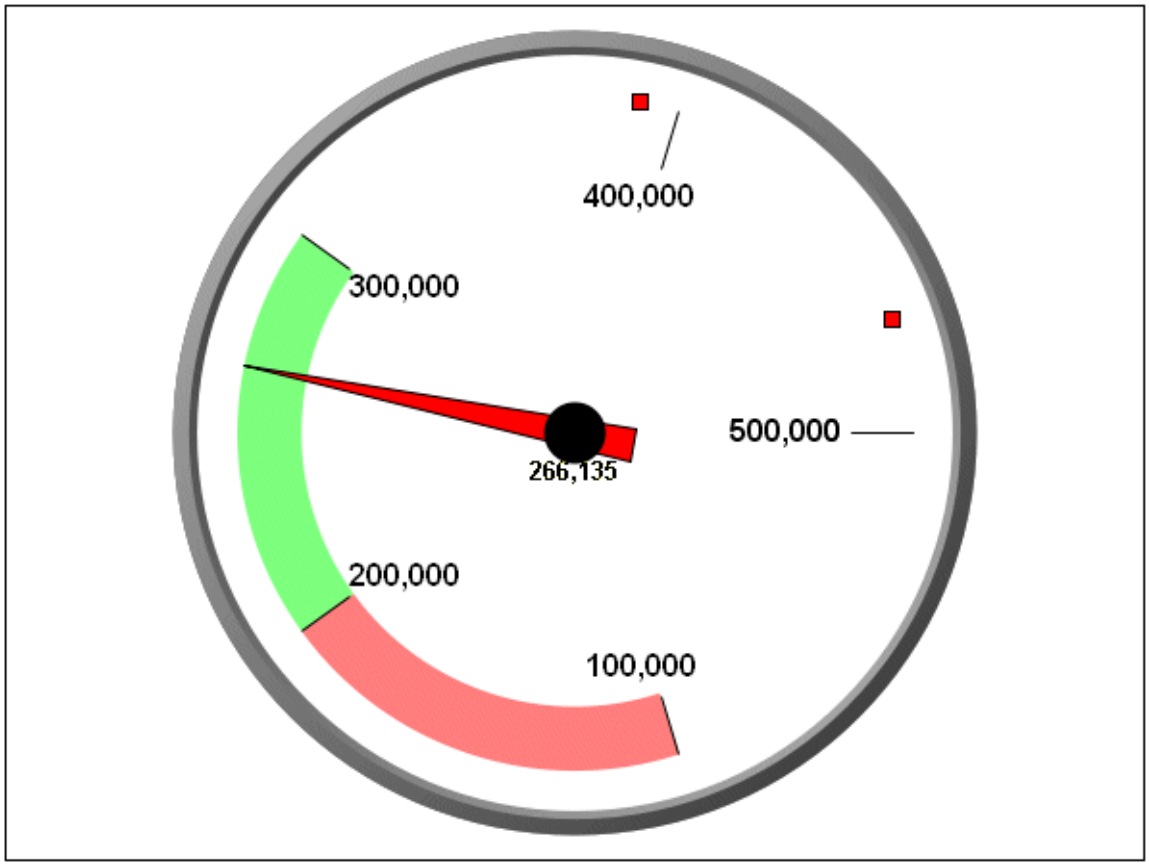
```
@GAUGE_BORDER_THICKNESS nThick
```

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 10
```



PERSISTENT:

NO

@GAUGE_CHART (Gauge Chart)

This macro creates a gauge chart.

SYNTAX:

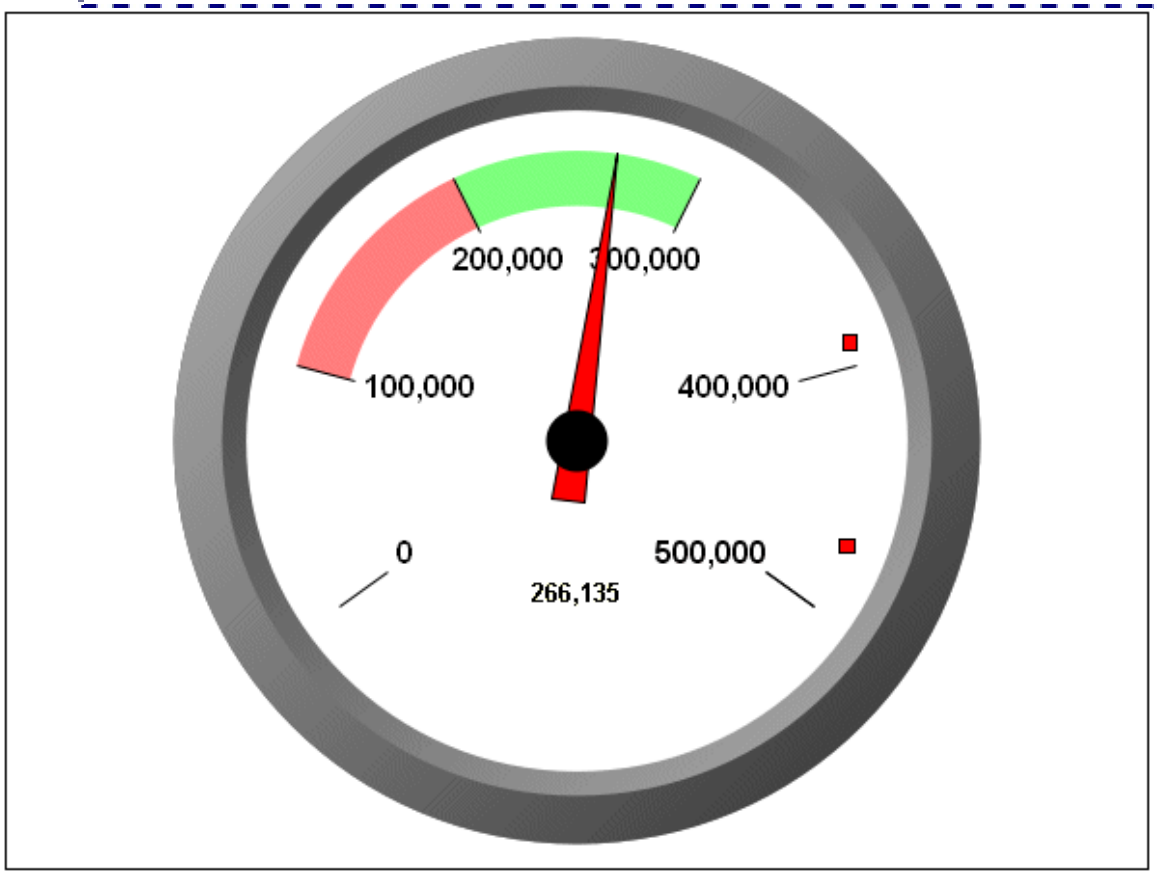
```
@GAUGE_CHART
```

PARAMETERS:

None

EXAMPLE:

```
@GAUGE_CHART
@SWAP 0
@GAUGE_MULTIPLE_NEEDLES 0
@GAUGE_BORDER_STYLE 5
@GAUGE_BORDER_THICKNESS 30
@GAUGE_RANGE_THICKNESS 20
@GAUGE_NEEDLE_STYLE 2
@GAUGE_RANGE_THICKNESS 30
@GAUGE_RANGE_COLOR 0 255 0 0 128
@GAUGE_RANGE_COLOR 1 0 255 0 128
@GAUGE_RANGE_START_STOP 0 100000 200000
@GAUGE_RANGE_START_STOP 1 200000 300000
```



PERSISTENT:

NO

@GAUGE_MULTIPLE_NEEDLES (Multiple Needles)

When a gauge chart includes more than one series, this macro controls the number of needles.

SYNTAX:

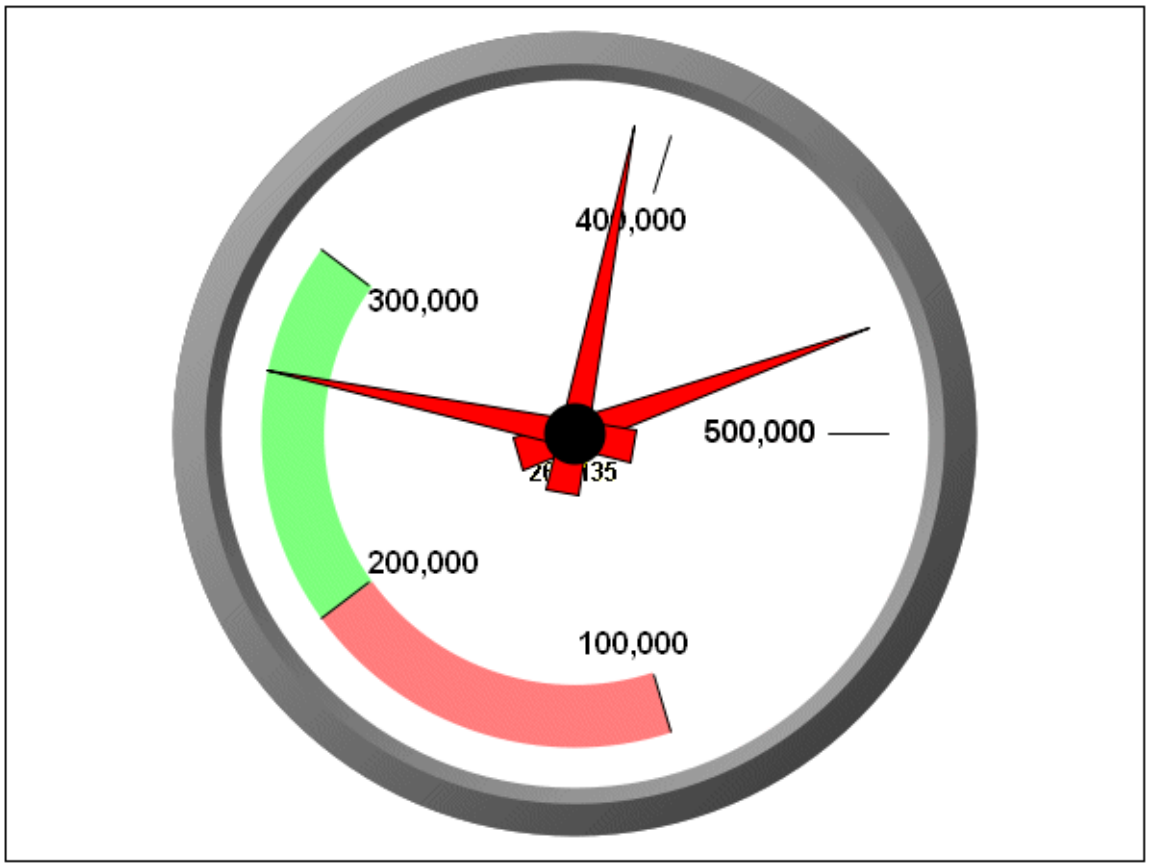
```
@GAUGE_MULTIPLE_NEEDLES bEnable
```

PARAMETERS:

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

EXAMPLE:

```
@GAUGE_MULTIPLE_NEEDLES 1
```



PERSISTENT:

NO

@GAUGE_NEEDLE_STYLE (Gauge Needle Style)

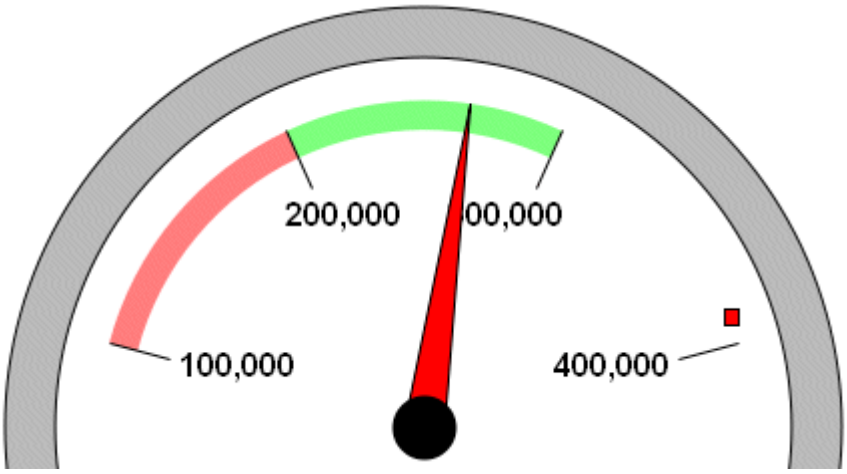
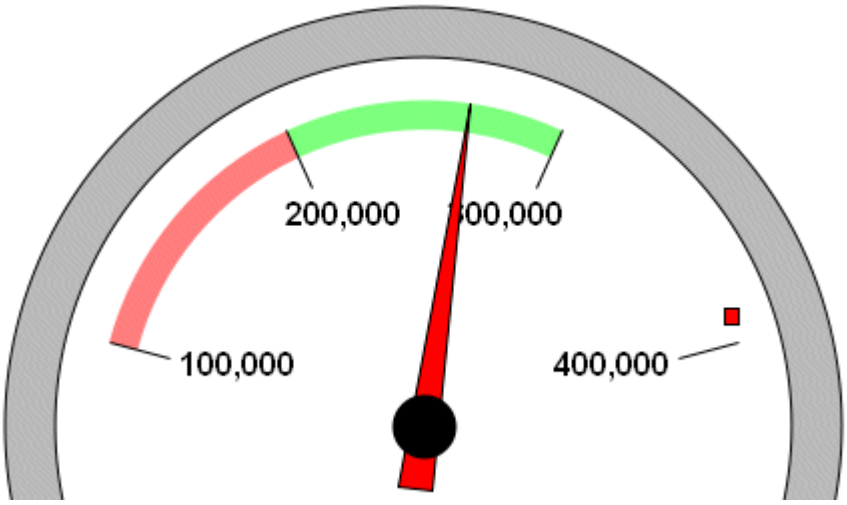
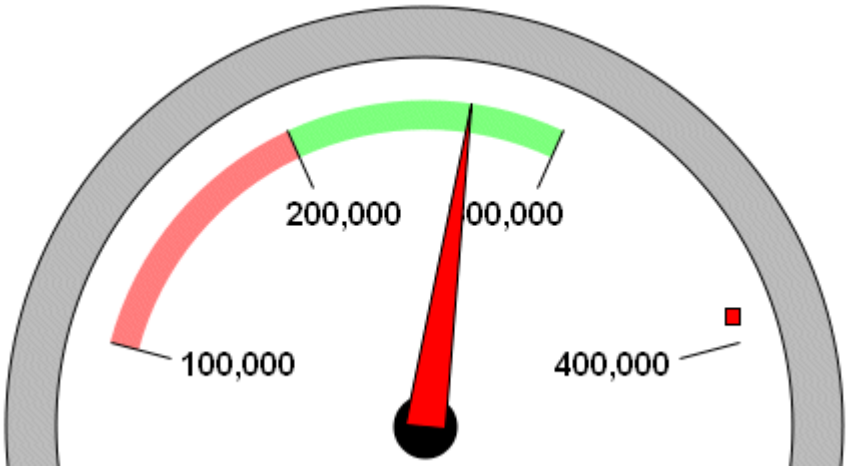
This macro can be used to select a needle style.

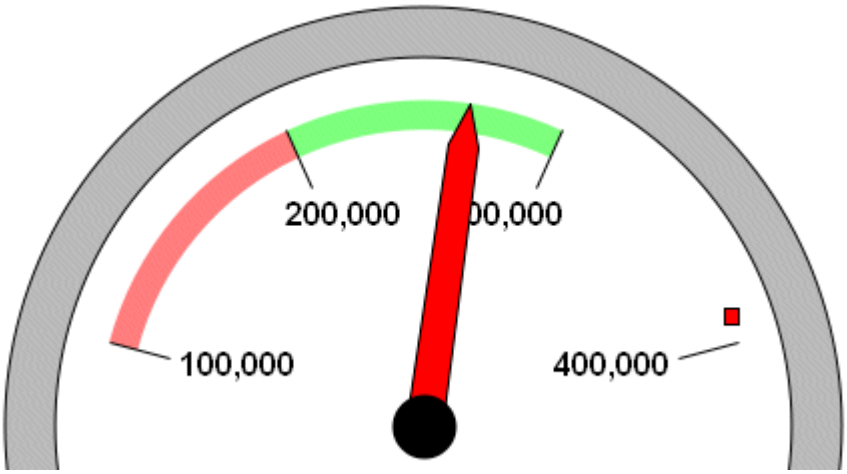
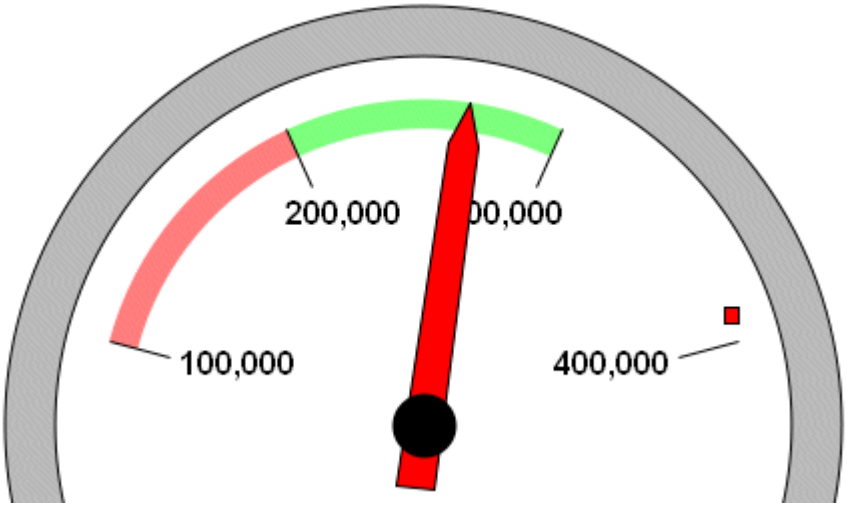
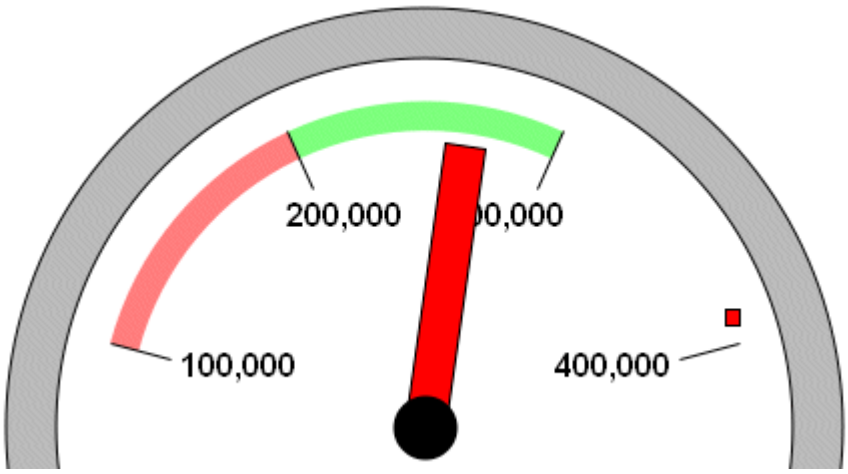
SYNTAX:

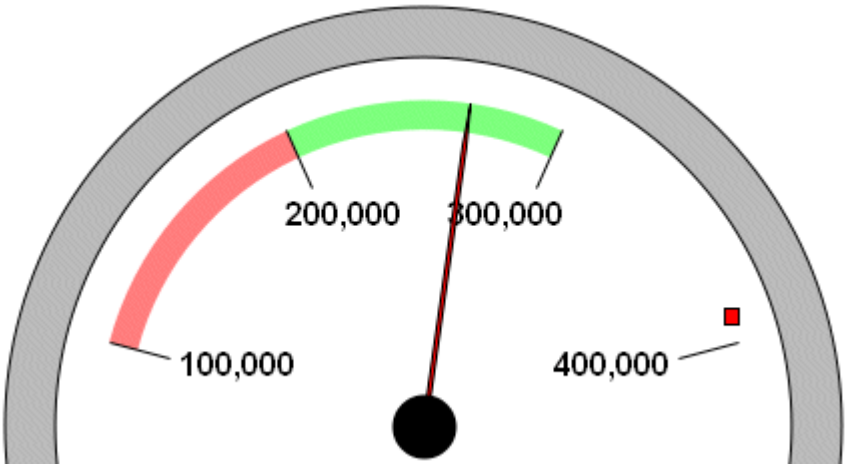
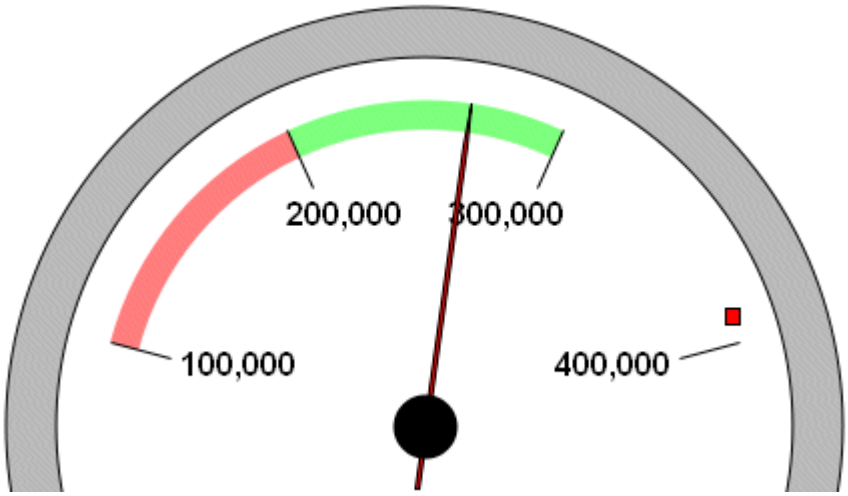
```
@GAUGE_NEEDLE_STYLE nStyle
```

PARAMETERS:

nStyle; 0...7 selects one of the following needle styles:

nStyle	Example	Description
0		Triangle
1		Triangle with extended end
2		Triangle above base

nStyle	Example	Description
3		Steeple
4		Steeple with extended end
5		Rectangle

nStyle	Example	Description
6		Thin
7		Thin with extended end

PERSISTENT:

NO

@GAUGE_RANGE_COLOR (Color Gauge Bands)

The @GAUGE_RANGE... macros can be used to draw up to five gauge bands on the face of the gauge. The @GAUGE_RANGE_START_STOP and @GAUGE_RANGE_THRESHOLD macros can be used to define the axis values at which the gauge bands start/end. The @GAUGE_RANGE_START_STOP macro defines the axis value where a band starts and ends. For example if the axis values in the gauge are 0...500,000, you might define a gauge band that starts at 100,000 and stops at 200,000 and another gauge band that starts at 200,000 and stops at 300,000. The @GAUGE_RANGE_THRESHOLD macro defines an ending band location. When this macro is used, the gauge band starts at the end of the previous band location (zero for gauge band zero). The @GAUGE_RANGE_COLOR macro can be used to assign a color to each gauge band. The @GAUGE_RANGE_THICKNESS macro defines the thickness of the band as a percentage of the available space inside the gauge. If @GAUGE_RANGE... macros are not defined, gauge bands are not drawn on the face of the gauge.

SYNTAX:

```
@GAUGE_RANGE_COLOR nBand nRed nGreen nBlue nAlpha
```

PARAMETERS:

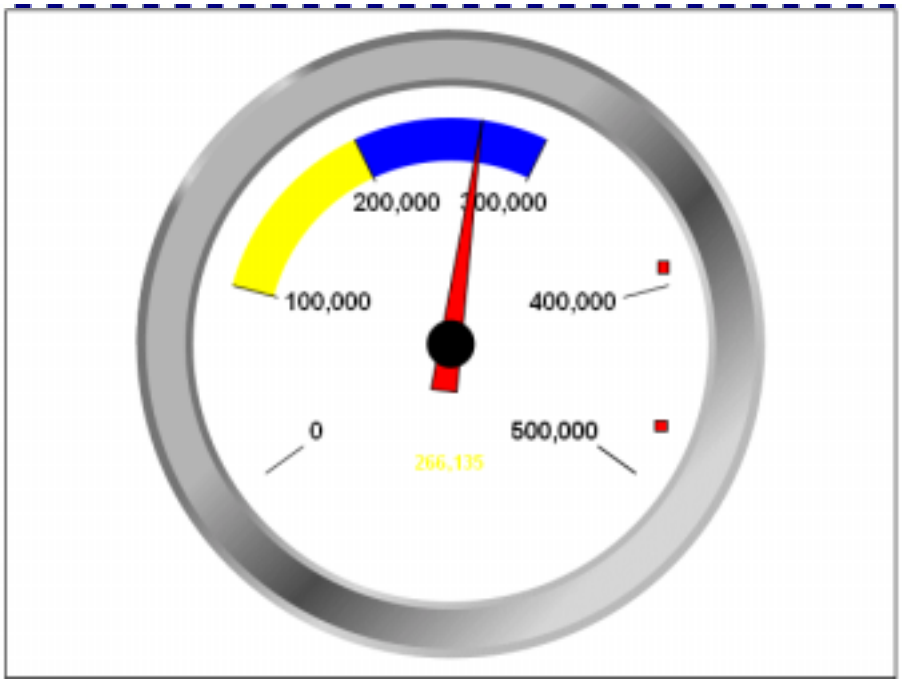
nBand; 0...5 selects the color band.

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

nAlpha; 0...255 selects the transparency level. 255 (the default) = no transparency. 0 = completely transparent

EXAMPLE:

```
@GAUGE_RANGE_COLOR 0 255 255 0 255
@GAUGE_RANGE_COLOR 1 0 0 255 255
@GAUGE_RANGE_START_STOP 0 100000 200000
@GAUGE_RANGE_START_STOP 1 200000 300000
```



PERSISTENT:

NO

@GAUGE_RANGE_START_STOP (Gauge Band Start/Stop)

The @GAUGE_RANGE... macros can be used to draw up to five gauge bands on the face of the gauge. The @GAUGE_RANGE_START_STOP and @GAUGE_RANGE_THRESHOLD macros can be used to define the axis values at which the gauge bands start/end. The @GAUGE_RANGE_START_STOP macro defines the axis value where a band starts and ends. For example if the axis values in the gauge are 0...500,000, you might define a gauge band that starts at 100,000 and stops at 200,000 and another gauge band that starts at 200,000 and stops at 300,000. The @GAUGE_RANGE_THRESHOLD macro defines an ending band location. When this macro is used, the gauge band starts at the end of the previous band location (zero for gauge band zero). The @GAUGE_RANGE_COLOR macro can be used to assign a color to each gauge band. The @GAUGE_RANGE_THICKNESS macro defines the thickness of the band as a percentage of the available space inside the gauge. If @GAUGE_RANGE... macros are not defined, gauge bands are not drawn on the face of the gauge.

SYNTAX:

```
@GAUGE_RANGE_START_STOP targetGaugeRange rangeStart rangeEnd
```

PARAMETERS:

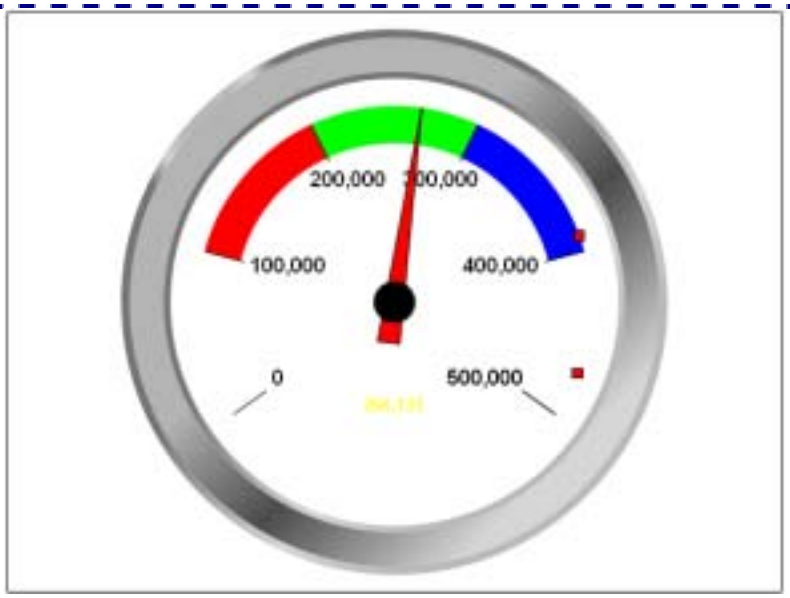
targetGaugeRange; 0...5 selects the color band.

rangeStart; Value at which to start *targetGaugeRange* band

rangeEnd; Value at which to end *targetGaugeRange* band

EXAMPLE:

```
@GAUGE_RANGE_COLOR 0 255 0 0 255
@GAUGE_RANGE_COLOR 1 0 255 0 255
@GAUGE_RANGE_COLOR 2 0 0 255 255
@GAUGE_RANGE_START_STOP 0 100000 200000
@GAUGE_RANGE_START_STOP 1 200000 300000
@GAUGE_RANGE_START_STOP 2 300000 400000
```



PERSISTENT:

NO

@GAUGE_RANGE_THICKNESS (Gauge Band Thickness)

The @GAUGE_RANGE... macros can be used to draw up to five gauge bands on the face of the gauge. The @GAUGE_RANGE_START_STOP and @GAUGE_RANGE_THRESHOLD macros can be used to define the axis values at which the gauge bands start/end. The @GAUGE_RANGE_START_STOP macro defines the axis value where a band starts and ends. For example if the axis values in the gauge are 0...500,000, you might define a gauge band that starts at 100,000 and stops at 200,000 and another gauge band that starts at 200,000 and stops at 300,000. The @GAUGE_RANGE_THRESHOLD macro defines an ending band location. When this macro is used, the gauge band starts at the end of the previous band location (zero for gauge band zero). The @GAUGE_RANGE_COLOR macro can be used to assign a color to each gauge band. The @GAUGE_RANGE_THICKNESS macro defines the thickness of the band as a percentage of the available space inside the gauge. If @GAUGE_RANGE... macros are not defined, gauge bands are not drawn on the face of the gauge.

SYNTAX:

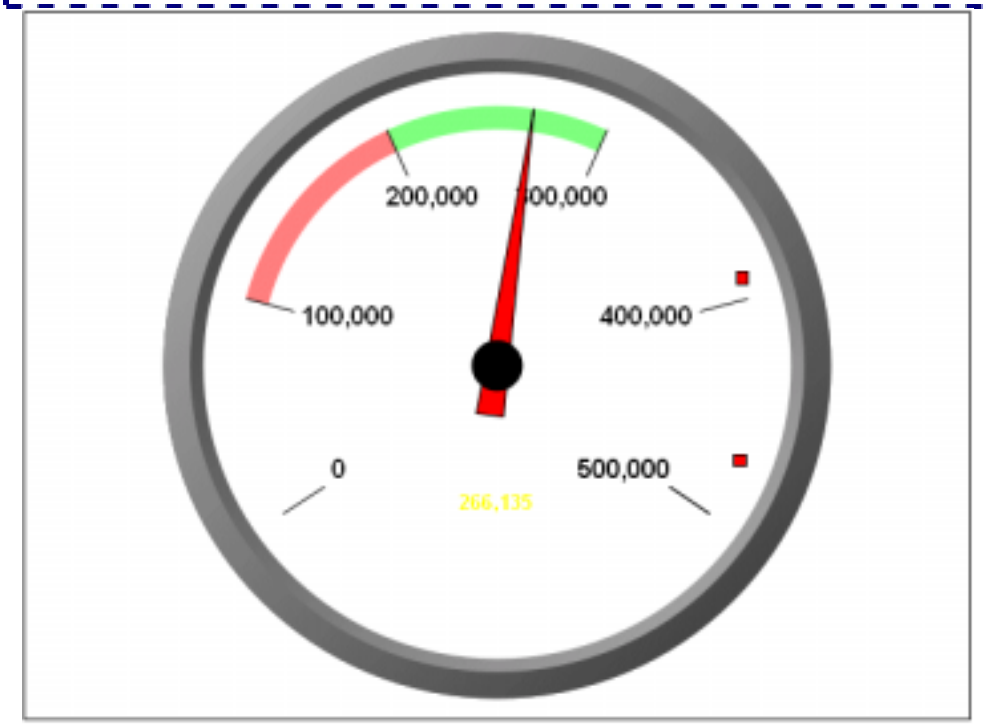
```
@GAUGE_RANGE_THICKNESS nThickness
```

PARAMETERS:

nThickness; 0...100 defines the thickness of the gauge bands.

EXAMPLE:

```
@GAUGE_RANGE_THICKNESS 10
```



PERSISTENT:

NO

@GAUGE_RANGE_THRESHOLD (Gauge Band Threshold)

The @GAUGE_RANGE... macros can be used to draw up to five gauge bands on the face of the gauge. The @GAUGE_RANGE_START_STOP and @GAUGE_RANGE_THRESHOLD macros can be used to define the axis values at which the gauge bands start/end. The @GAUGE_RANGE_START_STOP macro defines the axis value where a band starts and ends. For example if the axis values in the gauge are 0...500,000, you might define a gauge band that starts at 100,000 and stops at 200,000 and another gauge band that starts at 200,000 and stops at 300,000. The @GAUGE_RANGE_THRESHOLD macro defines an ending band location. When this macro is used, the gauge band starts at the end of the previous band location (zero for gauge band zero). The @GAUGE_RANGE_COLOR macro can be used to assign a color to each gauge band. The @GAUGE_RANGE_THICKNESS macro defines the thickness of the band as a percentage of the available space inside the gauge. If @GAUGE_RANGE... macros are not defined, gauge bands are not drawn on the face of the gauge.

SYNTAX:

```
@GAUGE_RANGE_THRESHOLD nBand fValue
```

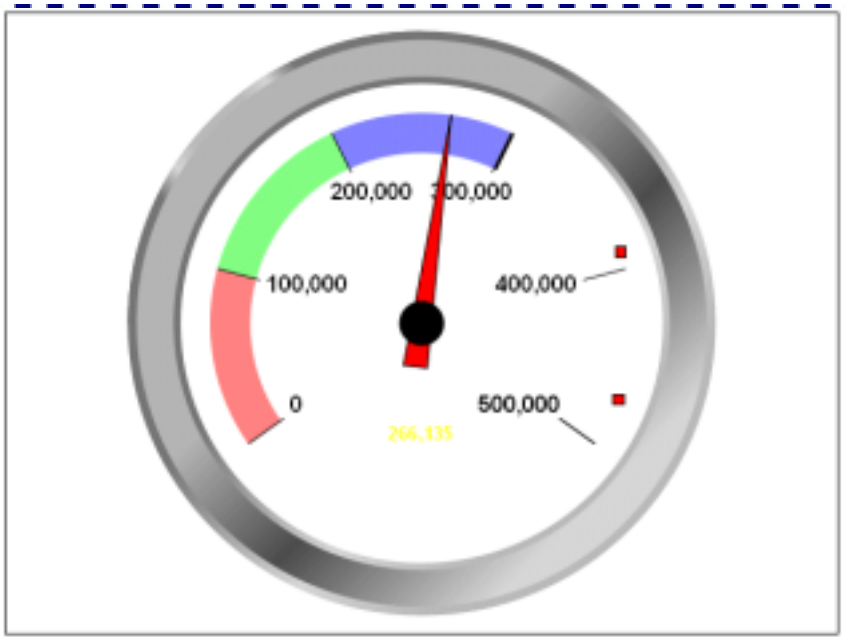
PARAMETERS:

nBand; 0...4 selects the color band

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

EXAMPLE:

```
@GAUGE_RANGE_THRESHOLD 0 100000
@GAUGE_RANGE_THRESHOLD 1 200000
@GAUGE_RANGE_THRESHOLD 2 300000
@GAUGE_RANGE_COLOR 0 255 0 0 125
@GAUGE_RANGE_COLOR 1 0 255 0 125
@GAUGE_RANGE_COLOR 2 0 0 255 125
```



PERSISTENT:

NO


THREE D GRAPHICS
CHARTS UNLIMITED

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
- @PIE_ROTATE; Set Pie Rotation Start Point
- @SMART_PIE_LABELS; Enable/Disable pie label layout engine

@DATATEXT_PIE (Data Text Mode for Pie Charts)

This macro sets the data text mode in pie charts.

SYNTAX:

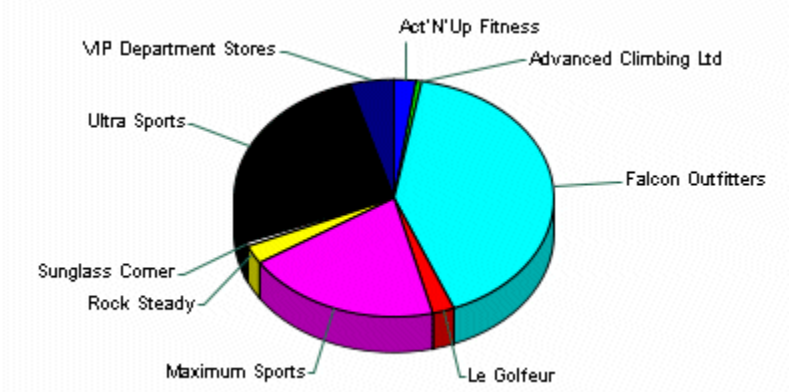
```
@DATATEXT_PIE nValue
```

PARAMETERS:

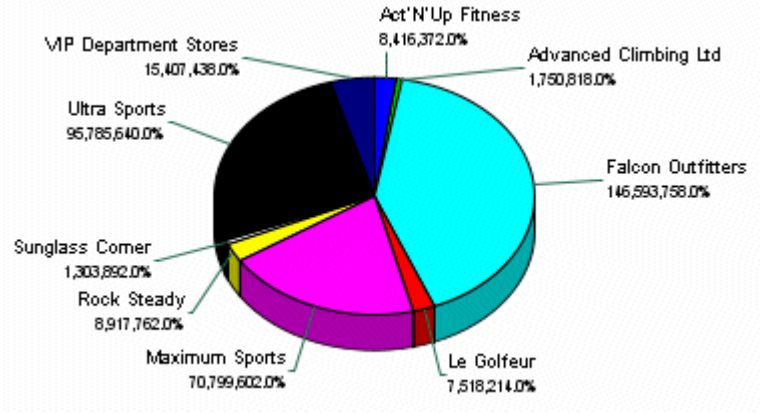
nValue; Selects the data text to show. It can be one of the following:
0 = NO Labels on Pie
1 = VALUE labels on Pie
2 = Series Labels on Pie
3 = Value and Series Labels on Pie

EXAMPLE:

```
@DATATEXT_PIE 2
```




```
@DATATEXT_PIE 3
```



PERSISTENT:

NO

@PIE_ROTATE (Pie Rotation Start Point)

This macro specifies the rotation start angle for pie charts.

SYNTAX:

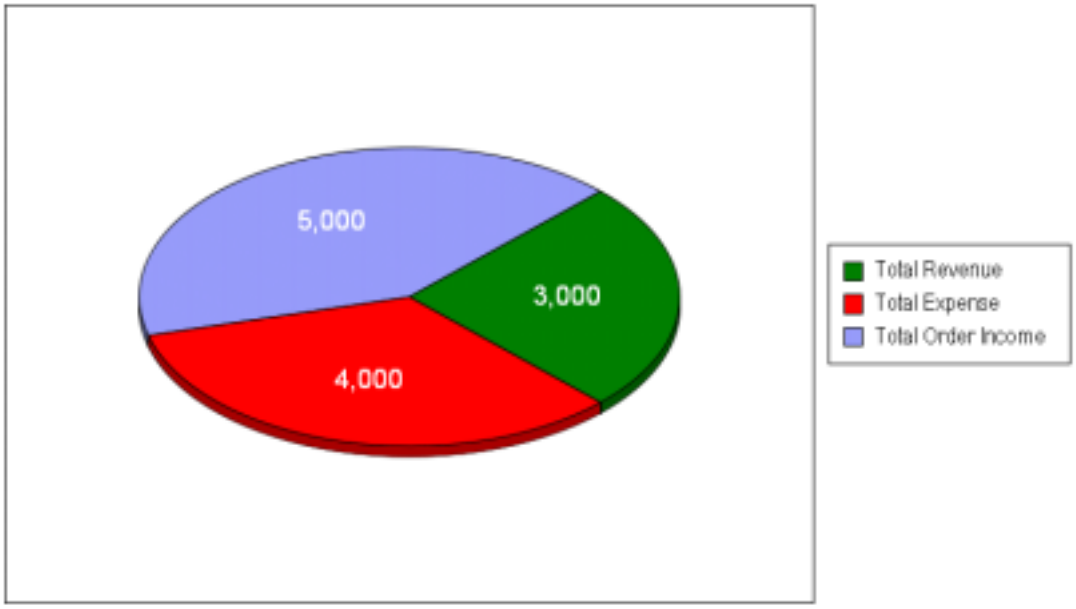
```
@PIE_ROTATE nRotation
```

PARAMETERS:

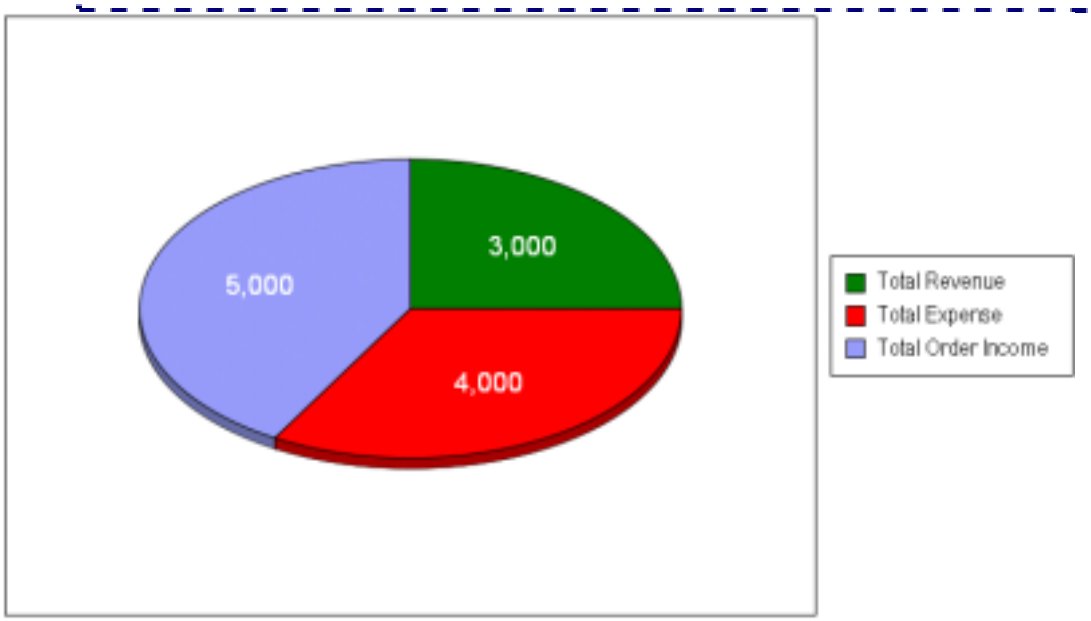
nRotation; 0...359 degrees to rotate pie

EXAMPLE:

```
@PIE_ROTATE 45
```




```
@PIE_ROTATE 90
```



PERSISTENT:

NO

@SMART_PIE_LABELS (Smart Pie Labels)

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

SYNTAX:

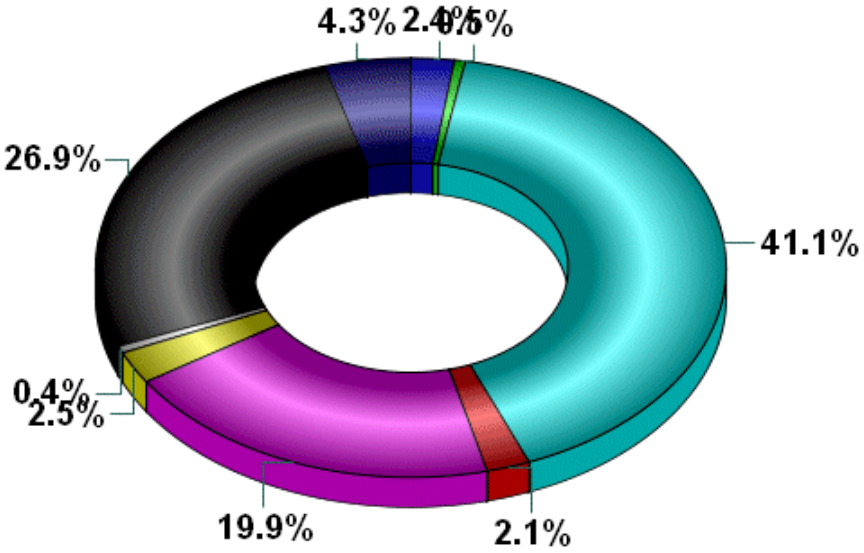
```
@SMART_PIE_LABELS bEnable
```

PARAMETERS:

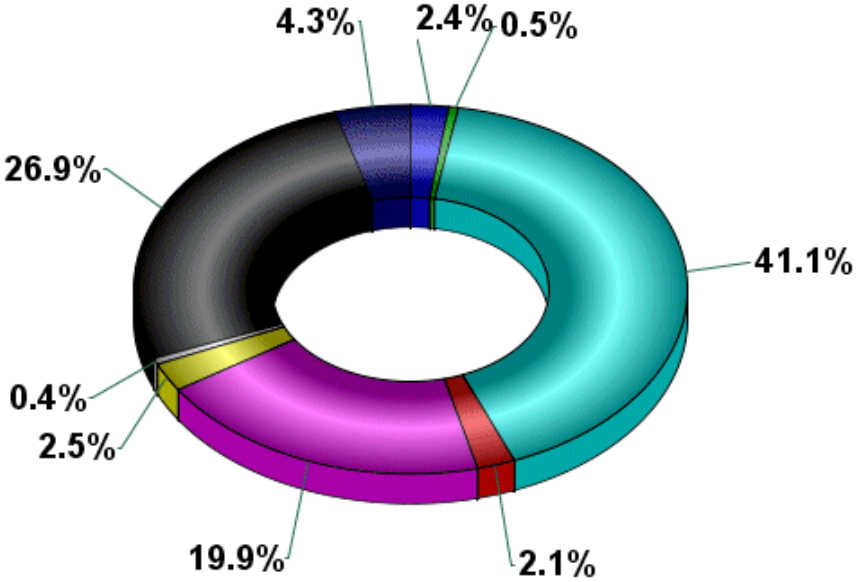
bEnable; 0=use standard pie engine, 1=use enhanced pie label layout engine

EXAMPLE:

```
@SMART_PIE_LABELS 0
```




```
@SMART_PIE_LABELS 1
```



PERSISTENT:

NO

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_CENTERTEXT; Center data text in a Waterfall Chart
- @WF_CONNECT; Assign a line style to feeler lines in a waterfall chart

@WATERFALL (Waterfall Chart)

This macro creates a waterfall chart.

SYNTAX:

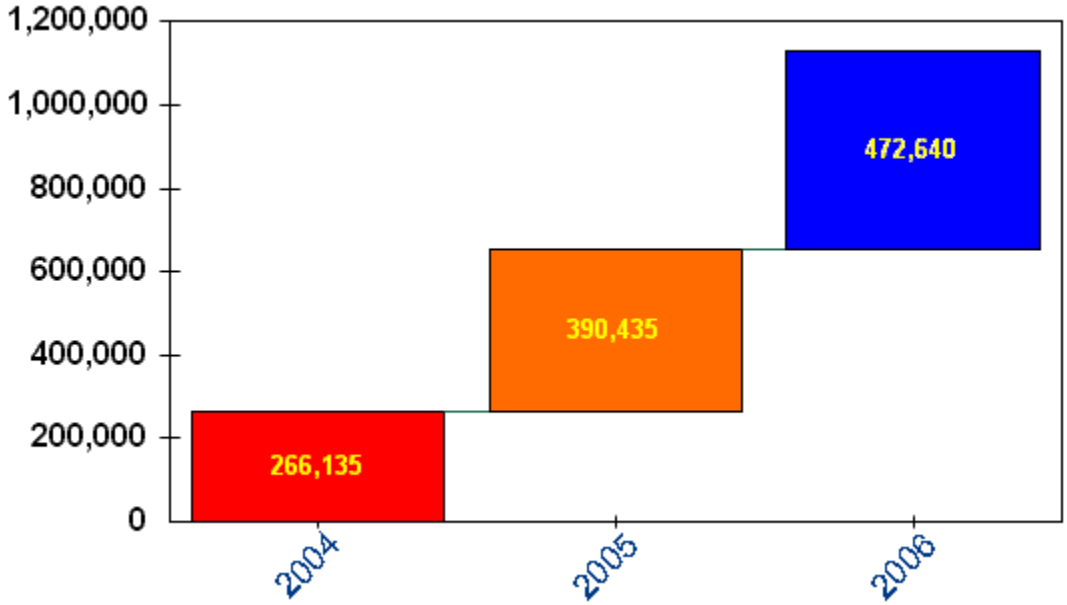
```
@WATERFALL
```

PARAMETERS:

None

EXAMPLE:

```
@WATERFALL
@WF_CONNECT 0
@WF_CENTERTEXT 1
```



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:

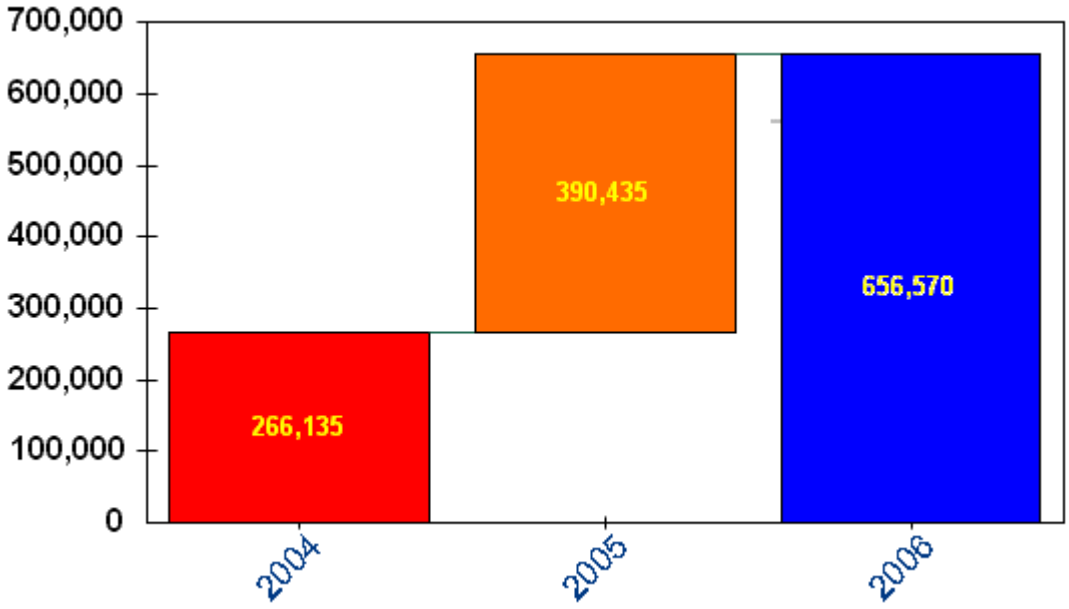
```
@WATERFALL2
```

PARAMETERS:

None

EXAMPLE:

```
@WATERFALL2
@WF_CONNECT 0
@WF_CENTERTEXT 1
```



PERSISTENT:

YES

ALSO SEE:

@WATERFALL

@WF_CENTERTEXT (Center Waterfall Data Text)

This macro specifies where data text is drawn in a waterfall chart. Set *bCenterText* to zero to draw data text normally below risers. Set *bCenterText* to one to draw data text in the center of risers.

SYNTAX:

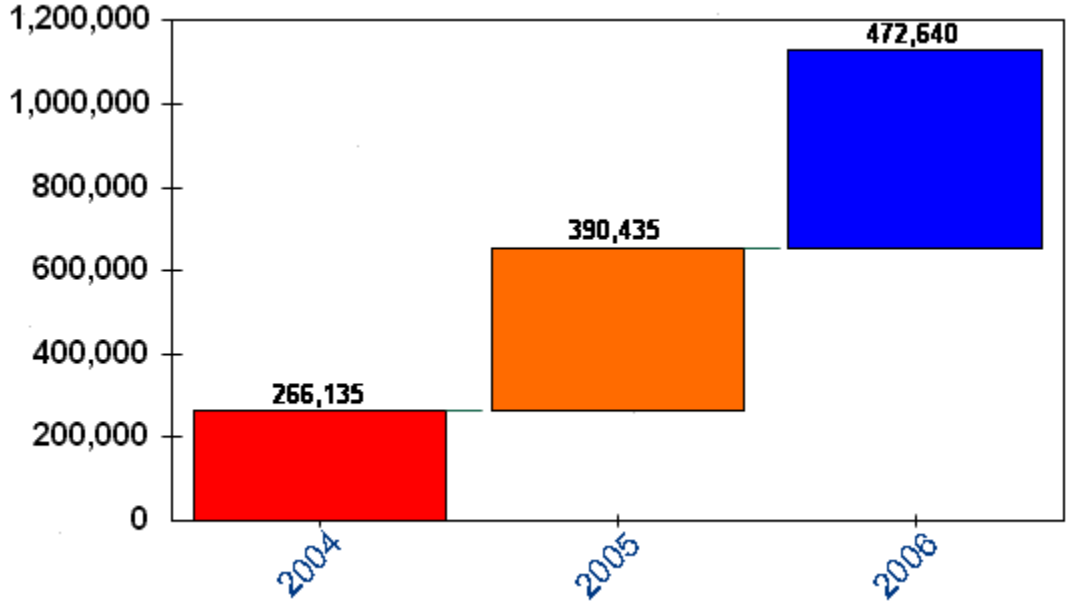
```
@WF_CENTERTEXT bCenterText
```

PARAMETERS:

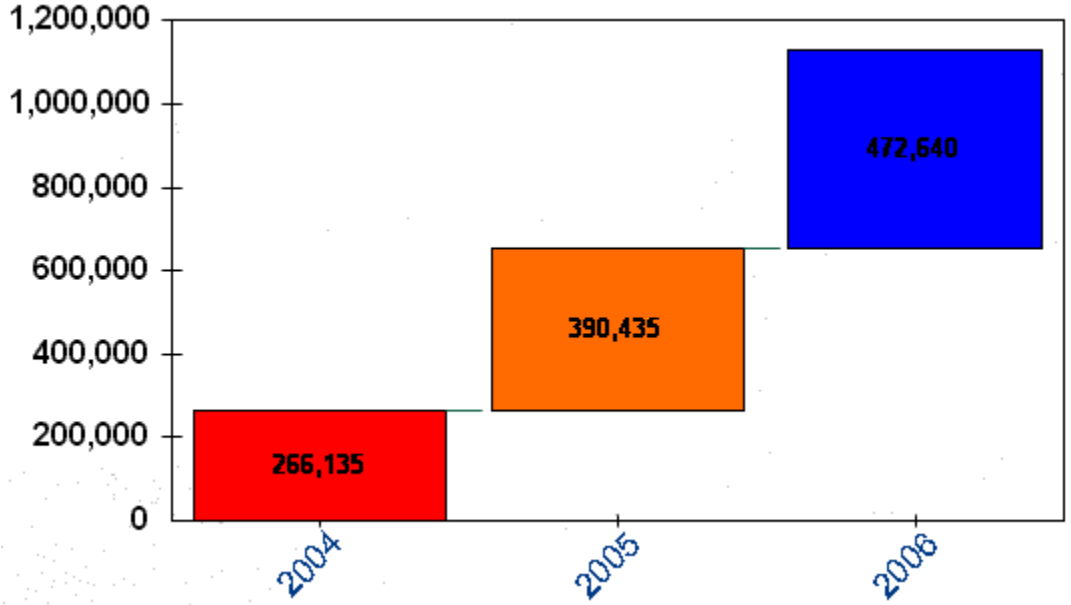
bCenterText; 1=Center data text, 0=Draw data text normally

EXAMPLE:

```
@WF_CENTERTEXT 0
```




```
@WF_CENTERTEXT 1
```



PERSISTENT:

NO

@WF_CONNECT (Waterfall Feeler Lines)

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

SYNTAX:

```
@WF_CONNECT nStyle
```

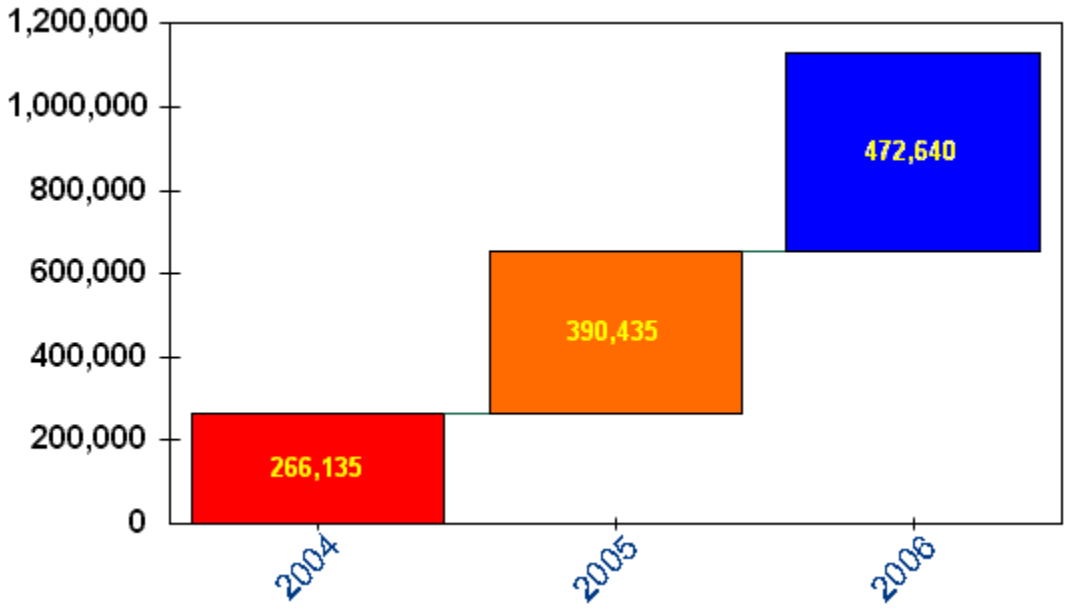
PARAMETERS:

nStyle; -1...15 selects one of the following line styles.

-1 = do not draw feeler lines
0 = Solid
1 = Dashed
2 = Dotted
3 = Dot-Dash
4 = Dash-Dot-Dot
5 = Medium Dash
6 = Short Dash
7 = Long Dash

EXAMPLE:

```
@WATERFALL
@WF_CONNECT 0
```



PERSISTENT:

YES


THREE D GRAPHICS
CHARTS UNLIMITED

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; Draw line chart to chart frame or inset
- @USER_LABEL_FONT; User-Defined Label Font

@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:

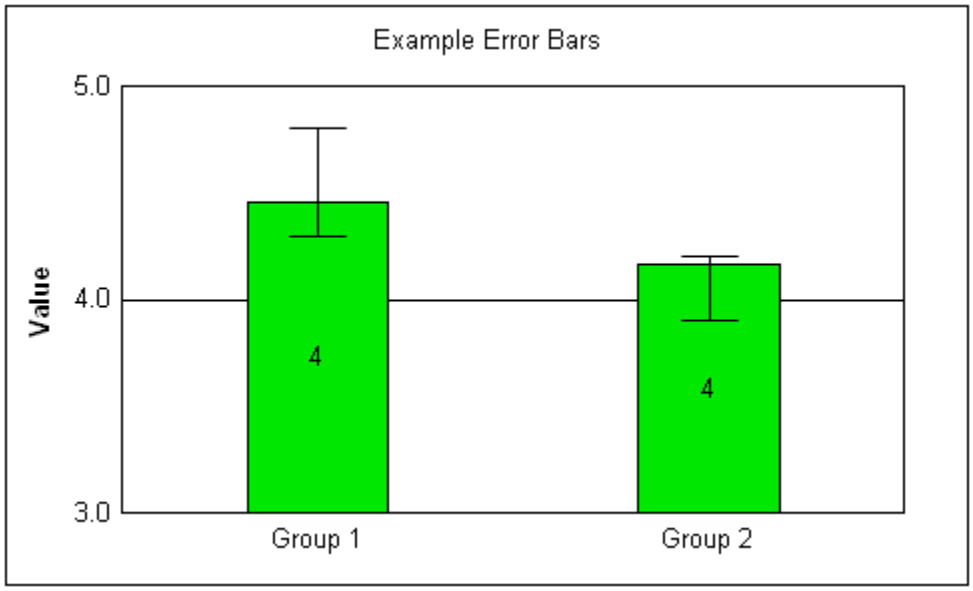
```
@EB n
```

PARAMETERS:

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

EXAMPLE:

```
@EB 1 @HAT 99
```



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
```

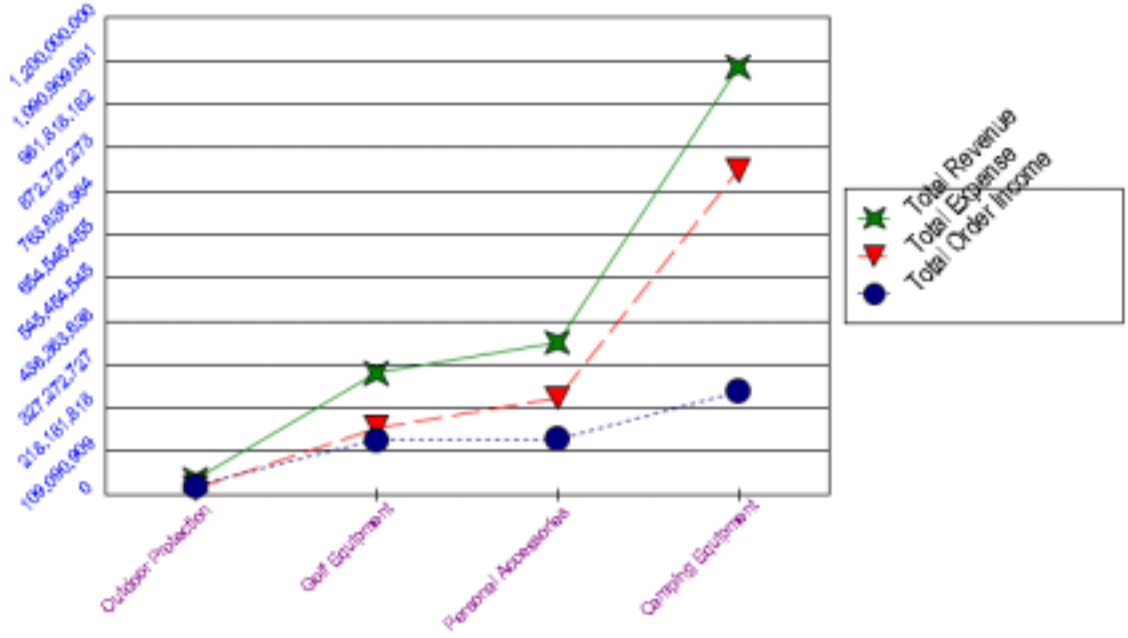
PARAMETERS:

nObject; 1...11 selects one of the following chart objects:
1 = Title
2 = Subtitle
3 = X1-Axis Title
4 = Y1-Axis Title
5 = Y2-Axis Title
6 = Footnote
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, 2...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 8 5
@FONTANGLE 9 5
@FONTANGLE 11 5
```



PERSISTENT:

YES

@FONTNAME (Font Name)

This macro can be used to assign a global font to a text 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 szFontName
```

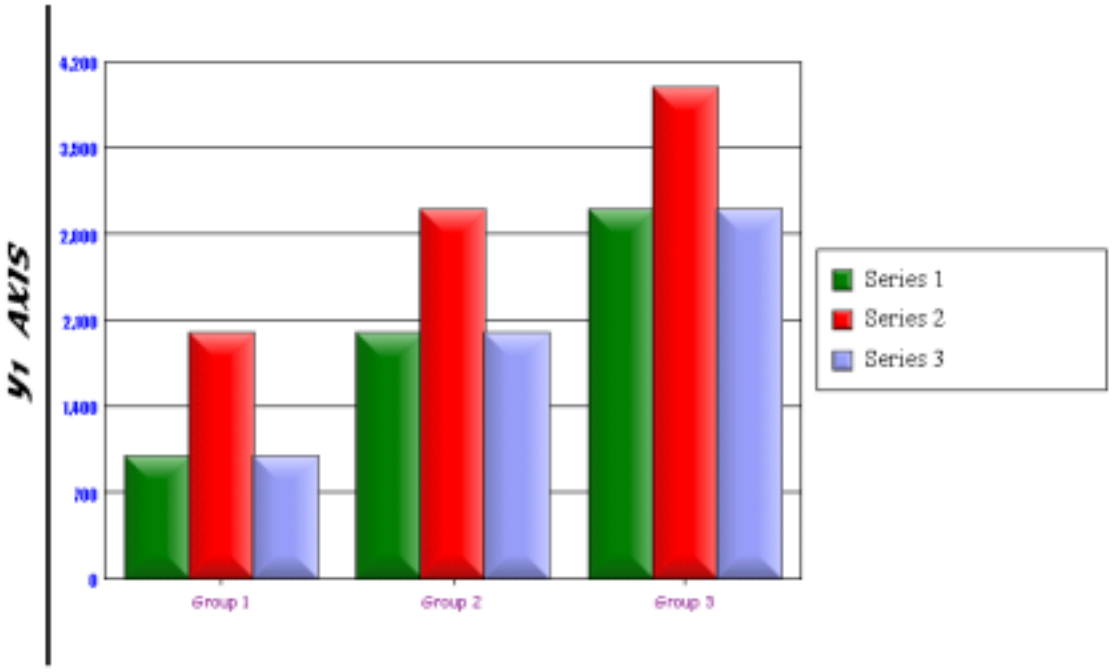
PARAMETERS:

nObject; 1...11 selects one of the following chart objects:
1 = Title
2 = Subtitle
3 = X1-Axis Title
4 = Y1-Axis Title
5 = Y2-Axis Title
6 = Footnote
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

szFontName; Font Name String (e.g., Times New Roman). Add a tilde character (~) to this string if you intend to define another macro in the same title field.

EXAMPLE:

```
@FONTNAME 4 Splash~ @FONTNAME 8 Comic Sans MS~ @FONTNAME 9 Impact~
@FONTNAME 11 Times New Roman
```



PERSISTENT:

NO

@FONTSIZE (Font Size)

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

SYNTAX:

```
@FONTSIZE nObject nSize
```

PARAMETERS:

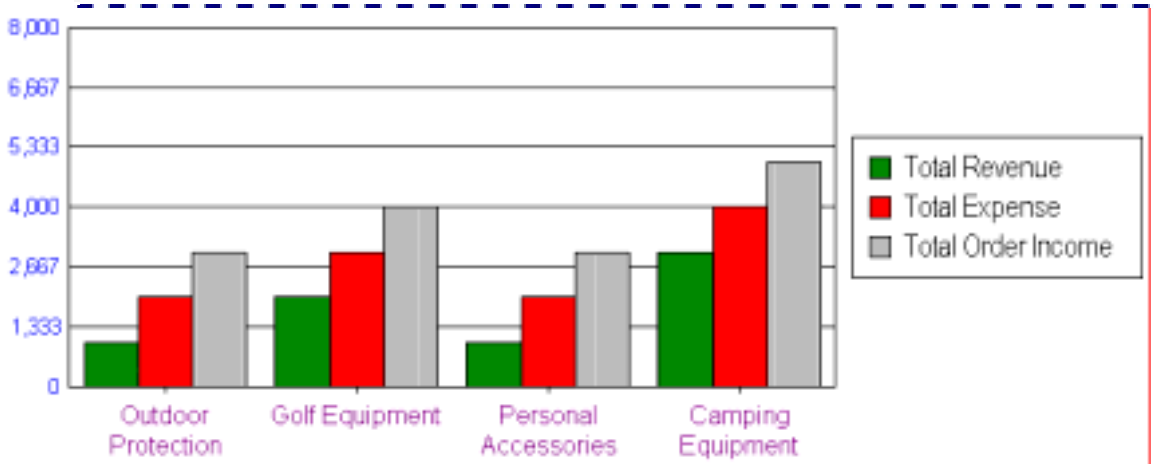
nObject; 1...11 selects one of the following chart objects:

1 = Title
2 = Subtitle
3 = X1-Axis Title
4 = Y1-Axis Title
5 = Y2-Axis Title
6 = Footnote
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:

```
@FONTSIZE 9 10
@FONTSIZE 8 12
@FONTSIZE 11 14
```



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
```

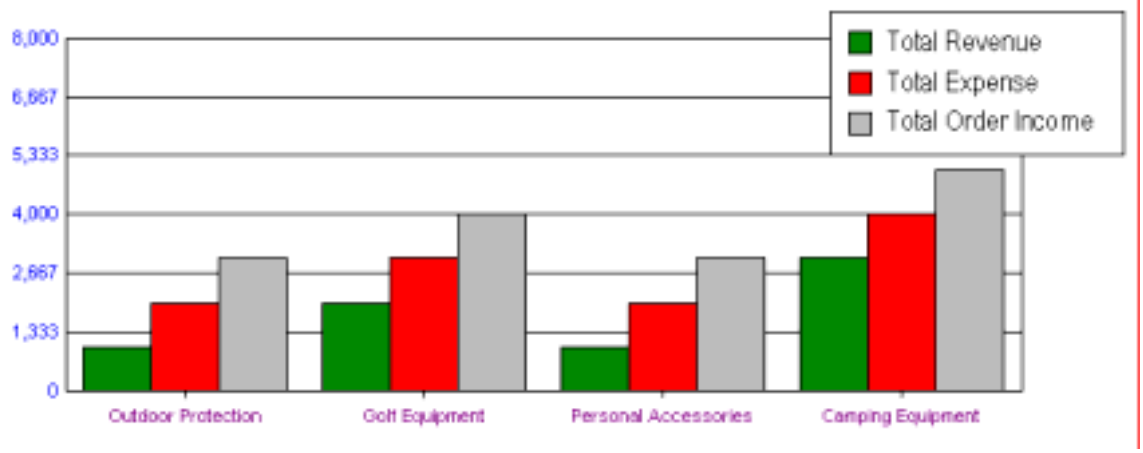
PARAMETERS:

nUpperLeftX, nUpperLeftY; -16383...+16383 specifies the X/Y position of the upper left corner of the chart frame in virtual coordinates

nLowerRightX, nLowerRightY; -16383...+16383 specifies the X/Y position of the lower right corner of the chart frame in virtual coordinates

EXAMPLE:

```
@FRAME -10000 1110 13383 -14000
```



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:

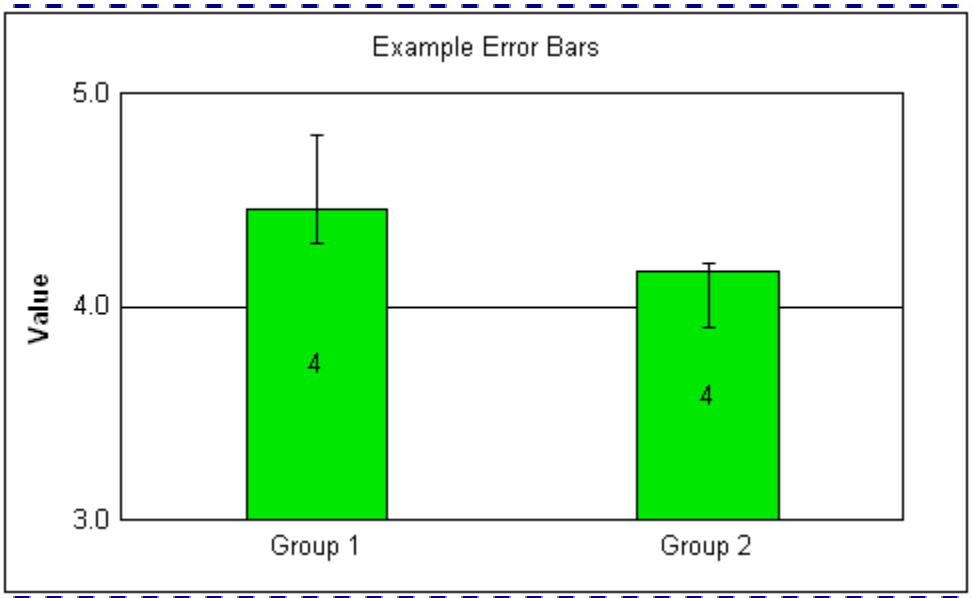
```
@HAT n
```

PARAMETERS:

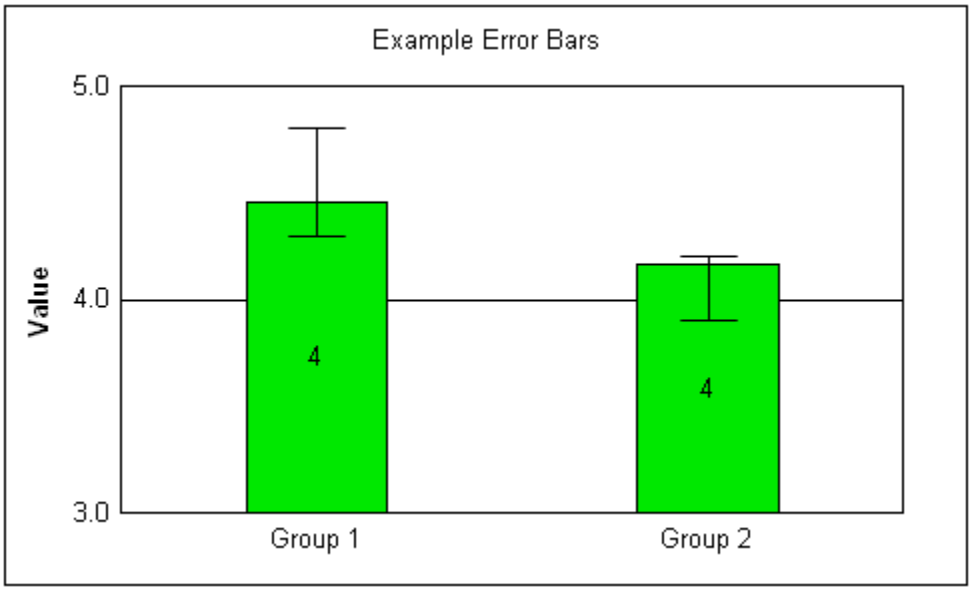
n; Width of Hat (0...100), 0=No Hat, 100=Widest Possible Hat.

EXAMPLE:

```
@EB 1 @HAT 20
```




```
@EB 1 @HAT 99
```



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
```

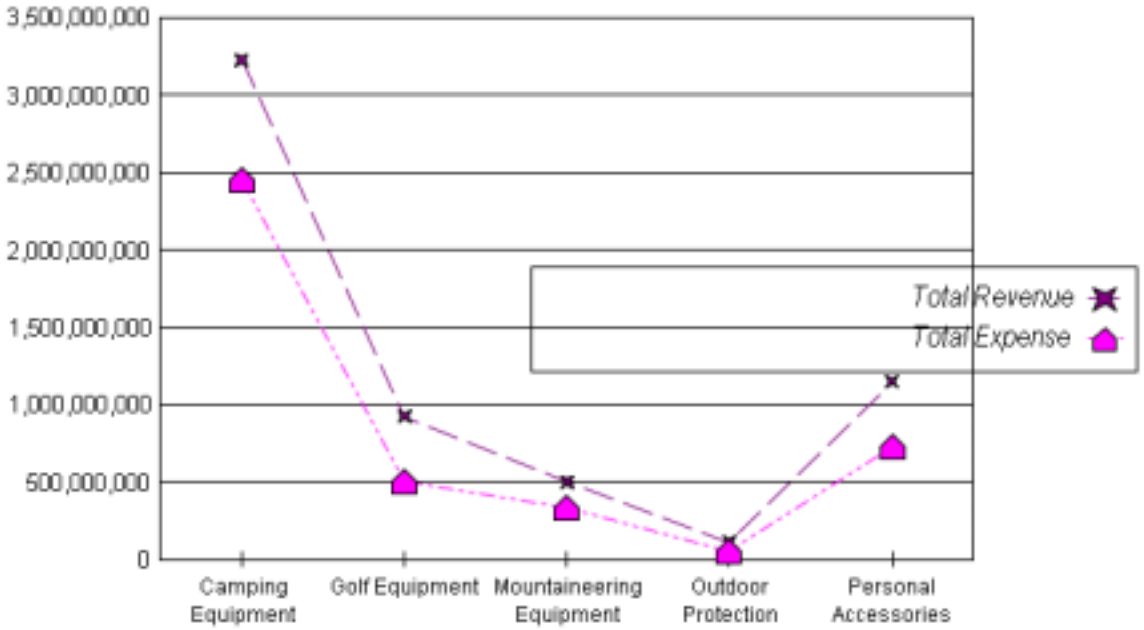
PARAMETERS:

nUpperLeftX, nUpperLeftY; -16383...+16383 specifies the X/Y position of the upper left corner of the chart legend in virtual coordinates

nLowerRightX, nLowerRightY; -16383...+16383 specifies the X/Y position of the lower right corner of the chart legend in virtual coordinates

EXAMPLE:

```
@LEGEND -1000 1110 13383 -1400
```



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:

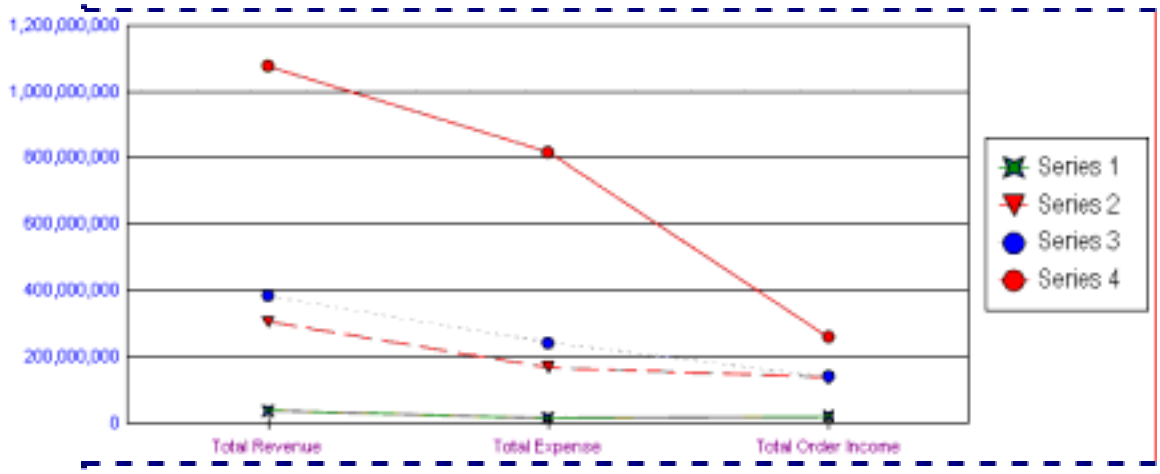
```
@LEGEND_ORDER nForceLegendOrder
```

PARAMETERS:

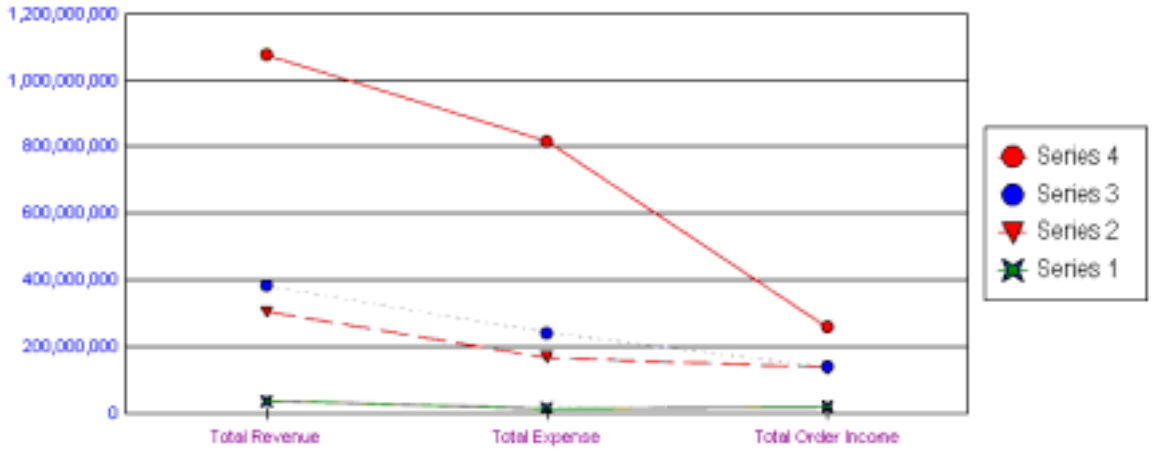
nForceLegendOrder; 0...2. 0 = Use internal logic to determine legend order, 1 = Force legend order to be Series 0.... Series N, 2 = Force legend order to be Series N .... Series 0

EXAMPLE:

```
@LEGEND_ORDER 1
```




```
@LEGEND_ORDER 2
```



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

PERSISTENT:

NO

@ORD_SPACE (Line Chart Drawing Mode)

This macro can be used to specify how line charts are drawn. Set *nMode* to zero for the normal mode of drawing line charts inset from the chart frame. Set *nMode* to one to extend the line chart to draw to the chart frame.

SYNTAX:

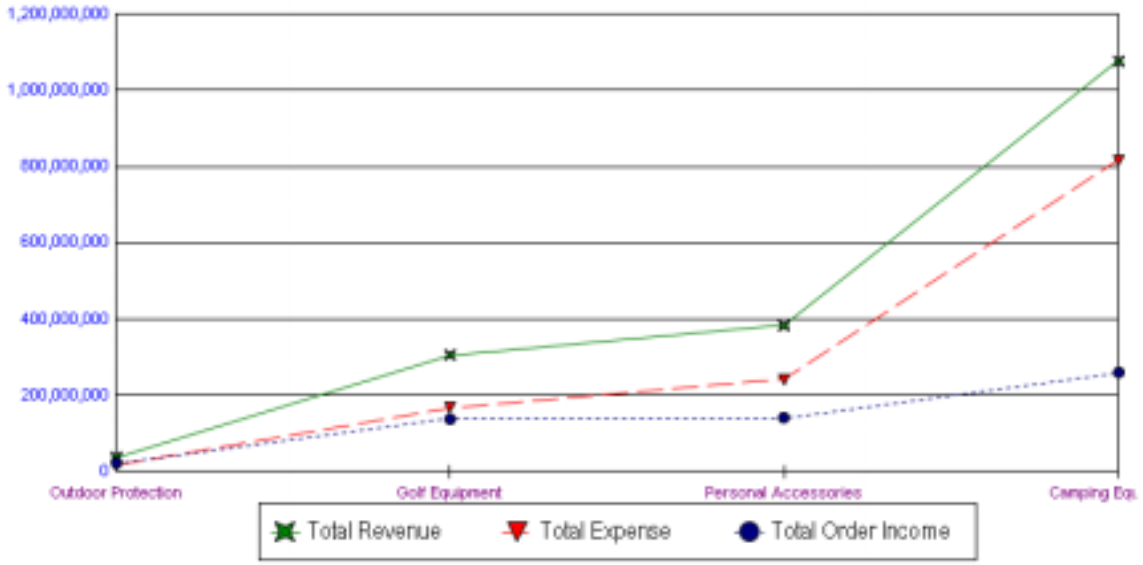
```
@ORD_SPACE nMode
```

PARAMETERS:

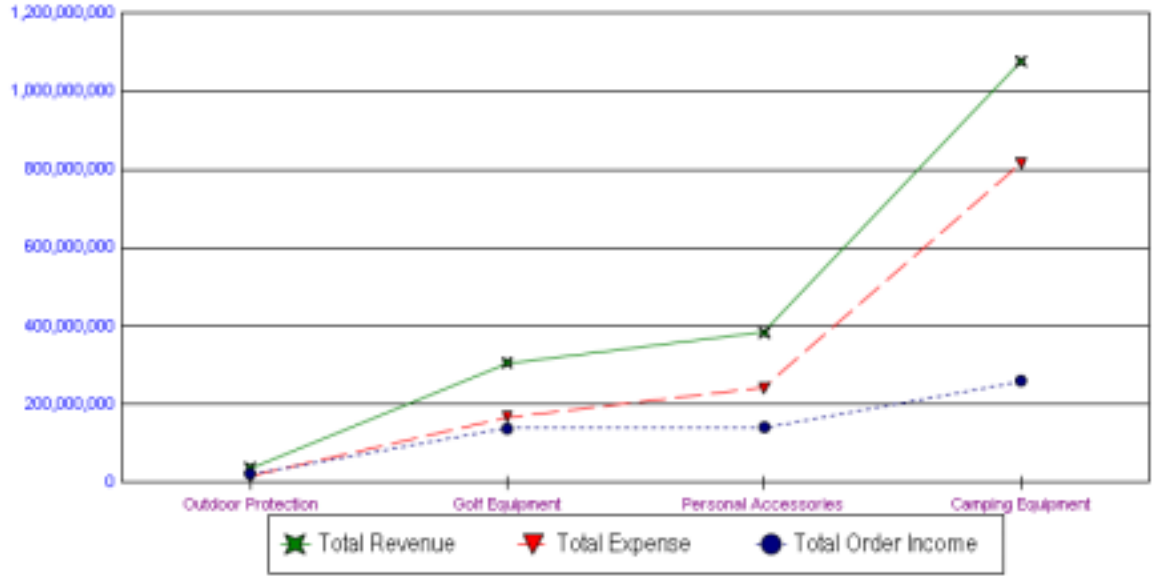
nMode; 0=draw line charts normal/inset, 1=draw line charts to chart frame

EXAMPLE:

```
@ORD_SPACE 1
```




```
@ORD_SPACE 0
```



PERSISTENT:

NO

@USER_LABEL_FONT (User-Defined Label Font)

This macro defines the size, color, and font to use for optional user-defined labels that are drawn with the following macros: @UF, @USER_CIRCLE, @USER_CIRCLE_ABOVE, @USER_FILL, @USER_FILL2, @USER_FILL_CIRCLE, @USER_FILL_CIRCLE_ABOVE, @USER_FILL_CIRCLE2, @USER_FILL_CIRCLE2_ABOVE, @USER_RECT, @UW, @XSZ, @XSZL, @XSZN, @XSZNL, @YSZ, @YSZL, @YSZN, & @YSZNL.

SYNTAX:

```
@USER_LABEL_FONT nSize nRed nGreen nBlue nAlpha szFont
```

PARAMETERS:

nSize; Font Point Size

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

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

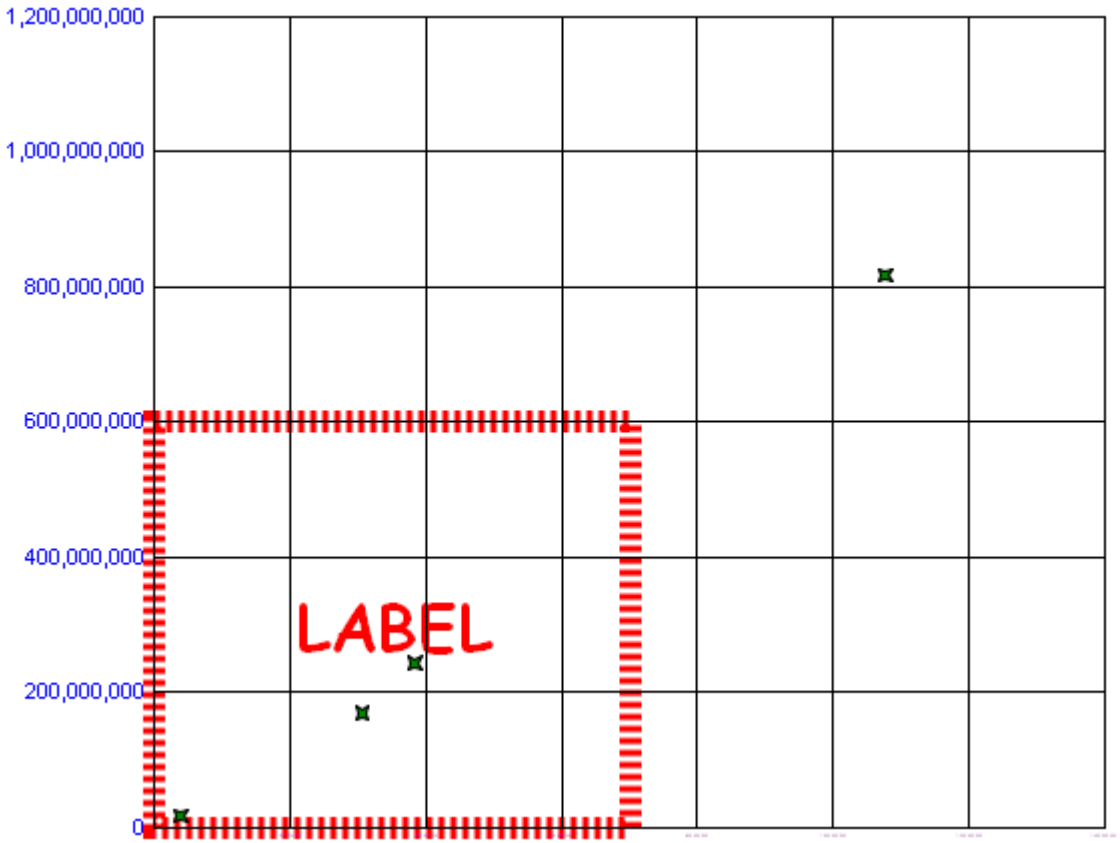
nBlue; 0...255 defines the BLUE portion of the RGB color

nAlpha; 0...255 selects the transparency level. 255 (the default) = no transparency. 0 = completely transparent.

szFont; Optional font name string (e.g., "Comic Sans MS").

EXAMPLE:

```
@USER_RECT 0.0 0.5 0.0 0.5 255 0 0 2 800 LABEL
@USER_LABEL_FONT 24 255 0 0 255 Comic Sans MS
```



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 Charts Unlimited. 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 Marco)

This macro provides useful information for tracking problems that may occur in Charts Unlimited. 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)

If you are using a field or function in Cognos as a parameter for a macro and do not achieve the expected results, this macro can be used to correct this problem. See "Using Fields/Functions in Macros" for more information about using fields and functions with Charts Unlimited macros. Normally, this macro is only needed in charts with a numeric X-axis.

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, 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

Appendix A: Alpha Index

Sec=Document Section where the macro is described

Macro	Description	Sec
@3DLABEL	Auto arrange 3D chart labels	5
@3DSCAT	Create a 3D Scatter Chart	11
@AA	Automatically arrange chart objects	2
@AA2	Automatically arrange chart objects with Frame adjustment.	2
@AA3	Automatically arrange chart objects with Frame & Legend adjustment.	2
@AA4	Automatically arrange chart objects with Maximized Frame adjustment.	2
@AGL	Alias/Change a Group Label	5
@ALPHA	Apply Alpha Channel Transparency to a riser or marker	8
@ALT_FRAME_FILL	Alternating Frame Fill	8
@ALT_FRAME_FILL_MODE	Alternating Frame Fill Mode	8
@APPEND_DATATEXT	Append a String to Data Text	7
@ASL	Alias/Change a Series Label	5
@ASL_DP	Map a sub-string from a label to the legend	5
@AUTO_COLOR	Automatic Color Mode	8
@AXIS	Assign a Series to an Axis	3
@BEVEL	Bevel effect chart objects	8
@BP/BP1	Box plot chart with a square-style tail	12
@BP2	Box plot chart with a T-style tail	12
@BP3	Box plot chart with a I-style tail	12
@BPH	Box plot chart orientation	12
@BPW	Box plot riser width	12
@BUBBLEMODE	Bubble Size in Bubble Charts	6
@CALC_PERCENT_SERIES	Calculate Percent Series	4
@COLOR_MODE	Color Mode (Color by Series or Group)	8
@COLOR_SCHEME	Activate a pre-defined color scheme	8
@COMBO	Create a Combo Chart	11
@COMPARE2	Create a 2-Series Absolute Bar Chart	11
@COND_COLOR & @COND_COLOR2	Color risers (bar/line/area) based on conditions	8
@COND_GROUP_LABEL	Color a riser based on a Group Label	5

Macro	Description	Sec
@COND_GROUP_LABEL2	Color a riser based on a Group Label prefix	5
@CURVED_LINES	Apply curved lines to a line or area chart	8
@CX	X1-Axis Line with Color	9
@CXY	Line between two X/Y Coordinates with Color	9
@CY	Y1-Axis Line with Color	9
@DATATEXT	Data Text Mode	7
@DATATEXT_PIE	Control appearance of pie chart data text	14
@DEBUG	Show Debug Information	17
@DEFINE_SCHEME	Define riser color scheme	8
@DLT	Data Line Type	7
@DP	Data Point Override	7
@DPC	Data Point Clear	7
@DT	Show/Hide Series-Specific Data Text	7
@DX	Number of X-Axis Divisions	3
@DY	Number of Y-Axis Divisions	3
@DY2	Number of Y2-Axis Divisions	3
@EB	Error bars in standard bar/column charts	16
@FONTANGLE	Font Rotation angle of a text object	16
@FONTNAME	Map a global font onto a font object	16
@FONTSIZE	Font size of a text object	16
@FORCE_ABSOLUTE	Force Series to plot Absolute	4
@FORCE_DATALINE	Force dataline connecting all datapoints	7
@FORCE_DATATEXT_CURRENCY	Force data text to currency format	7
@FORCE_SERIES_COUNT	Force Series Count	4
@FORECAST	Add Blank Groups	4
@FRAME	Define frame size/location	16
@GANTT	Create a Gantt Chart	11
@GAUGE_ANGLE	Start/Stop Angle of Gauge labels/gridlines	13
@GAUGE_BORDER_STYLE	Gauge border	13
@GAUGE_BORDER_THICKNESS	Gauge border thickness	13
@GAUGE_CHART	Create a Gauge Chart	13
@GAUGE_MULTIPLE_NEEDLES	Gauge multiple needles	13
@GAUGE_NEEDLE_STYLE	Gauge needle style	13

Macro	Description	Sec
@GAUGE_RANGE_COLOR	Gauge band colors	13
@GAUGE_RANGE_START_STOP	Location on gauge axis where band starts/stops	13
@GAUGE_RANGE_THICKNESS	Thickness of internal gauge bands	13
@GAUGE_RANGE_THRESHOLD	Gauge Band Thresholds	13
@GCOLOR	Color a Chart Object	8
@GM	Read data in Column or Row Major order	4
@GRAPHTYPE	Graph Type	11
@GROUP_LABELS_ON_BASELINE	Group Labels on Baseline	5
@GX	Grid Style for the X-Axis	3
@GY	Grid Style for the Y-Axis	3
@HAT	Size of hats on Error Bars	16
@HIDE_ZERO	Hide Risers/Markers for zero data points	6
@HL	Highlight a Riser/Marker	6
@HOUR_SCALE	Minor axis HOUR scale on a time axis	3
@IG	Ignore a Group	4
@IN	Move First Box Plot In/Right	12
@INIT_USERLINES	Initialize user-defined lines	9
@IR	Insert Row	7
@IS	Ignore a Series	4
@LEGEND	Define legend size/location	16
@LEGEND_ORDER	Force Legend Order	16
@LEGEND_WRAP_WIDTH	Virtual wrap point for legend text	16
@LIMIT_VISIBLE_GROUPS	Limit Visible Groups	4
@LS	Line Style	9
@MARKER	Define Marker Shapes	6
@MC	Marker Colors in Box Plots	12
@MCOLOR	Define Marker Colors	6
@MEAN	Mean average line	9
@MIN_GROUPS	Add blank groups to control bar size	4
@MK	Number of Markers in Box Plots	12
@MOVA	Standard/scientific Moving Average Line	9
@MS	Marker Shapes in Box Plots	12
@OFFSCALE_Y1	Offscale Values on the Y1-Axis	7
@ORD_SPACE	Draw line charts to frame edge or inset	16
@PARAM_COUNT_FIXUP	Debug Macro	17

Macro	Description	Sec
@PARAM_FIXUP	Fix parameters from Crystal Reports Fields/Functions.	17
@PARETO	Create a Pareto Chart	11
@PAT	Apply Pattern to Riser	6
@PERCENT_SERIES	Create Ratio Series	4
@PIE_ROTATE	Pie rotation start point	14
@POLAR	Create a Polar Chart	11
@PX	X-Axis Precision	3
@PY	Y1-Axis Precision	3
@PY2	Y2-Axis Precision	3
@RDT	Rotate Data Text	7
@RESET	Reset Internal Data Range	17
@RG	Reverse Groups	4
@RISER_BORDER	Enable/Disable Riser Borders	6
@RISER_OVERLAP	Overlap between risers in side-by-side/clustered bar charts	6
@RISER_WIDTH	Riser Width in bar charts	6
@RIVER	Create floating area (river) from 2 series	11
@RS	Reverse Series	4
@SC	Define Y-Axis Scale	3
@SCX	Define X-Axis Scale	3
@SCY2	Define Y2-Axis Scale	3
@SHADOW	Apply a Drop Shadow	8
@SINGLE_GROUP	Draw only the first Group in a Scatter Chart	4
@SMART_PIE_LABELS	Enable/Disable pie label layout engine	14
@SMOOTH_LINE	Connect data points with Smooth or Straight Line segments	9
@SORT	Sort Series/Groups	4
@STEP_LINE	Series Stepped Line	4
@STEP_LINE2	Series Stepped Line at Values	4
@SWAP	Swap Series/Groups	4
@SZ	Size of Markers in Scatter or Box Plot charts	6
@TOTAL_GROUP	Create a Total Group	4
@TRENDLINE	Trend Line across a series	9
@TRENDLINE_ALLDATA	Linear Regression Line through all data	9

Macro	Description	Sec
	points	
@UF	Same as @USER_FILL	10
@USER_CIRCLE	User-Defined Outlined Circle	10
@USER_CIRCLE_ABOVE	User-Defined Outlined Circle above the Chart Area	10
@USER_FILL	User-Defined Color-Filled Rectangle	10
@USER_FILL_CIRCLE	User-Defined Color-Filled Circle	10
@USER_FILL_CIRCLE_ABOVE	User-Defined Color-Filled Circle above the Chart Area	10
@USER_FILL_CIRCLE2	User-Defined Pattern-Filled Circle	10
@USER_FILL_CIRCLE2_ABOVE	User-Defined Pattern-Filled Circle Above the Chart Area	10
@USER_FILL2	User-Defined Pattern-Filled Rectangle	10
@USER_LABEL_FONT	User-Defined Label Font	16
@USER_MARKER	User-Defined Marker at X/Y coordinates	10
@USER_MARKER2	User-Defined Marker at X/Y coordinates with Value	10
@USER_RECT	User-Defined Outlined Rectangle	10
@USER_SERIES	User-Defined Series	4
@UW	User-Defined Vertical Band	10
@WATERFALL	Create a normal Waterfall Chart	15
@WATERFALL2	Create a Waterfall Chart with Total Group	15
@WC	Color an @UW Vertical Band	10
@WF_CENTERTEXT	Center Data Text in a Waterfall Chart	15
@WF_CONNECT	Line style of feeler lines in a waterfall chart	15
@X	X-Axis Line at Value	9
@X_AXIS_MODE	X-Axis Mode	3
@XG	X-Axis Line at Group	9
@XSKIP	Skip labels on X-Axis	5
@XSKIP2	Skip labels on X-Axis/Force Last Label	5
@XSKIP3	Max X-labels/auto-adjust skip to match	5
@XSZ	X-Axis Line with Label	9
@XSZL	X-Axis Line with Label on Left	9
@XSZN	X-Axis Line with Label & Value	9
@XSZNL	X-Axis Line with Label & Value on Left	9

Macro	Description	Sec
@XY	X/Y Coordinates Line	9
@XY_DP2	Line between two Data Points (Scatter Charts)	9
@Y	Y-Axis Line	9
@Y_ZERO	Include/Exclude Zero for Auto Scale	7
@Y1_FORCE_PERCENT	Y1-Axis Percent Format	3
@Y1_FORCE_PERCENT2	Y1-Axis Percent/Numeric Format	3
@Y1_INVERT	Invert the Y1-Axis Invert	3
@Y1BASE	Y1-Axis Base Line	3
@Y2_FORCE_PERCENT	Y2-Axis use Percent Format	3
@Y2_INVERT	Y2-Axis Invert	3
@Y2BASE	Y2-Axis Base Line	3
@Y2SLAVE	Slave Y2-Axis to Y1-Axis	3
@Y2SLAVE2	Slave Y1/Y2-axes to Max Value	3
@YSZ	Y-Axis Line with Label	9
@YSZL	Y-Axis Line with Label on Left	9
@YSZN	Y-Axis Line with Label & Value	9
@YSZNL	Y-Axis Line with Label & Value on Left	9


THREE D GRAPHICS
CHARTS UNLIMITED