THREE D GRAPHICS

CRCHART

(Chart Enhancement Macros for Crystal Reports)

Version: 2.37.3

May, 2005

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

Telephone: 1.310.231.3330 | Fax: 1.310.231.3303 | Email: info@threedgraphics.com

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

Email: info@threedgraphics.com

Web: http://www.threedgraphics.com

April, 2005

Table of Contents

CRCHART

Overview

CRChart is a replacement charting library for Crystal Reports. It adds significant new charting capabilities for Crystal Reports designers and developers. This "replacement library" is 100% compatible with your Crystal Decisions® product. It simply expands your charting and graphing capabilities. Features:

- Box Plots
- 3D Scatter Charts
- User programmable lines on any axis (or free floating)
- User programmable markers at any point on the chart
- Error Bars
- Pivot functions (Swap series/group, data reversal, etc...)
- Conditional Coloring and Formatting
- Runtime editing functions for most chart properties
- Drop Shadow and Alpha Channel 'transparency' effects
- Ability to use any Crystal Reports field or function as an input parameter to CRChart macros
- and much more!

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 2000, XP, NT, or ME
- Crystal Reports 10.x, 9.x or 8.x

Note that these versions of Crystal Reports are included in some versions of Microsoft Visual Basic 6 and Visual Studio.NET.

Installation & Setup

Before you begin installation, make a backup copy of the existing charting library (sscsdk80.dll). The following table shows where this file is stored for each supported version of Crystal Reports:

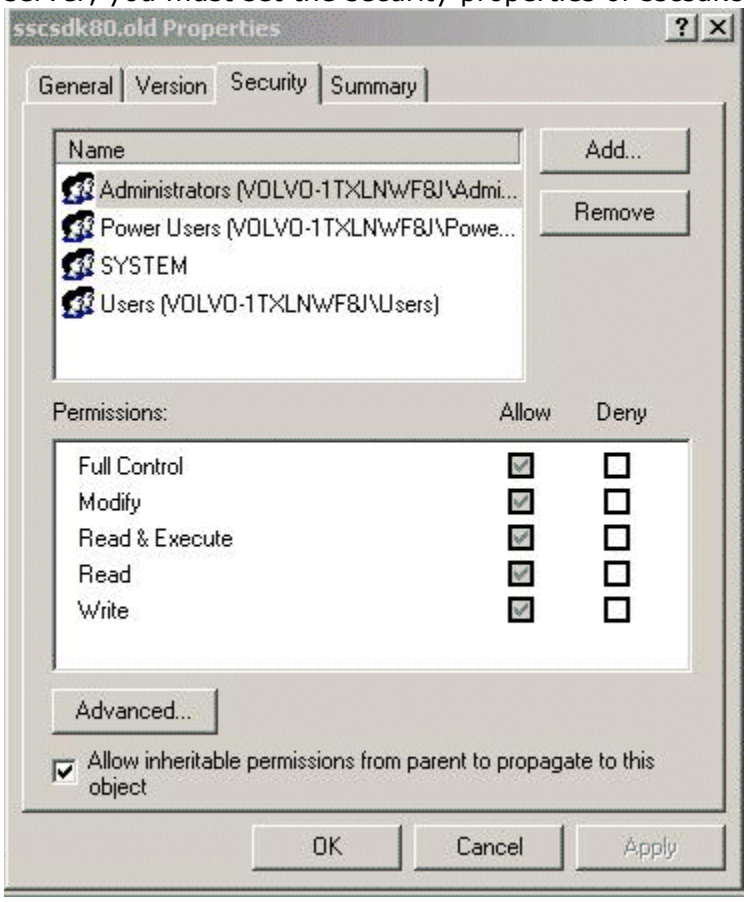
Version	Path
10.x	C:\Program Files\Common Files\Crystal decisions\2.5\bin
9.x	C:\Program Files\Common Files\Crystal Decisions\2.0\bin
8.x	C:\Program Files\Seagate Software\Shared
Run-time version packaged with .Net	C:\Program Files\Common Files\Crystal Decisions\1.0\bin

Use one of the following installation methods:

- Run the CRChart installation wizard to install the CRChart library (sscsdk80.dll) on your system.
- Manual Installation. Copy the provided CRChart library (sscsdk80.dll) to the location indicated in the table.

NOTE: *For developers who are preparing applications for distribution that use the charting library with these special chart enhancement macros, the sscsdk80.dll file must be shipped with your application.*

If CRChart is installed with Crystal Reports 10.x and .NET running on a server, you must set the security properties of sscsdk80.dll.

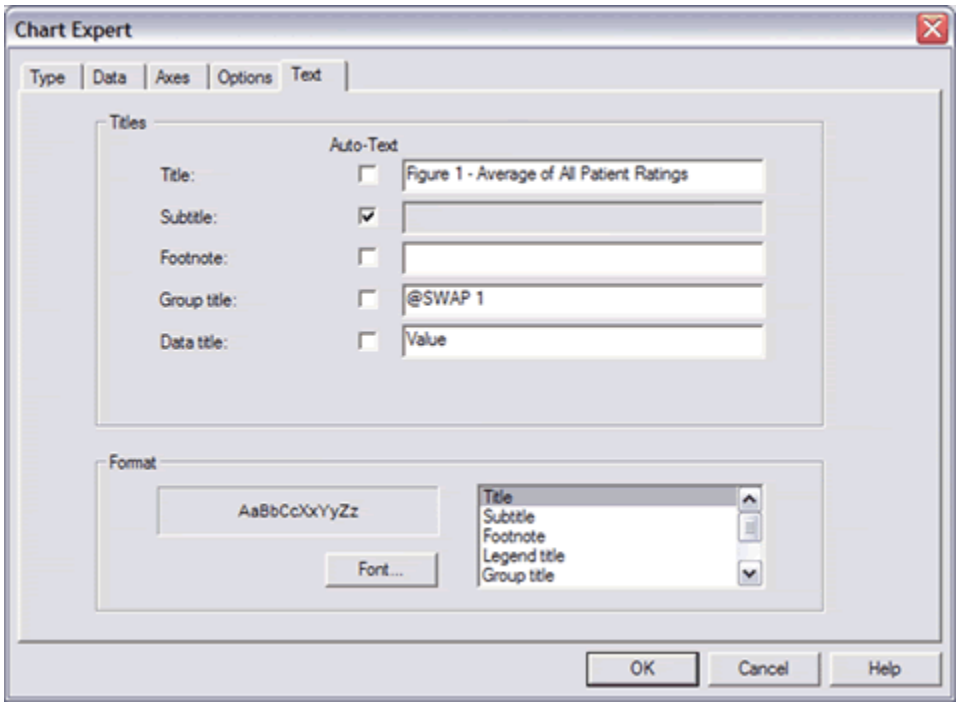


- Check the "Allow inheritable permission from parent to propagate to this object" checkbox.
- Click the OK button
- Restart the .NET server.

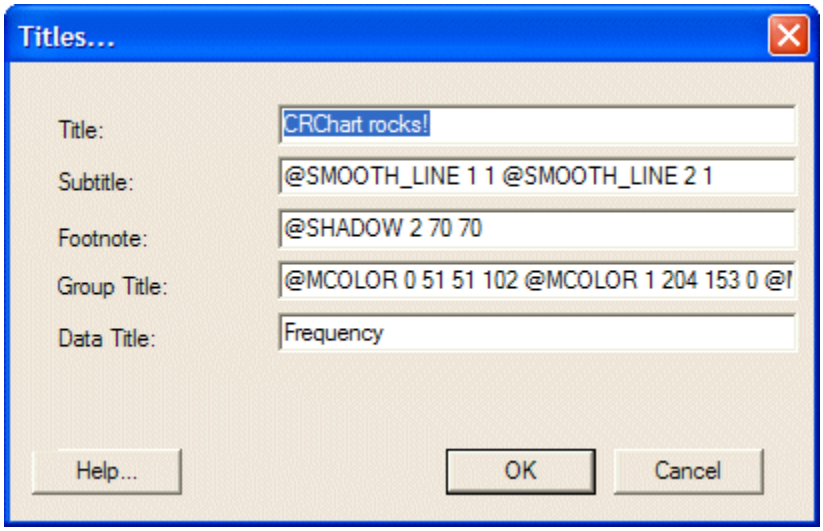
General Usage Notes

CRChart macros can be inserted in any chart title field (i.e., title, subtitle, footnote, etc.) field in a Crystal Reports chart. Chart titles can be defined in the Text tab section of the Chart Expert dialog or in the Titles dialog that is available from the Chart Options menu.

- Right-click on the chart background and choose Chart Expert to show the Chart Expert dialog.
- Select the Text tab to show chart title fields.



- Right click on the chart background and choose Chart Options to shown the Chart Options menu.
- Choose Titles from the Chart Options menu to show the 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 tilda (~) character and a space to the title and add the macro after the space. The space after the tilda is important. If it is not included, the macro will be ignored.

Example:

```
TitleText~ @3DSCAT
```

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

Example:

```
@USER_SERIES 2 3.6 3.8  Foo~
```

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
```

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: -1...*n* (where: *n* is the total number of series in the chart). Minus one (-1) is a special value that will apply the macro to all series in the chart. A value of zero (0) selects series 1.

Using Crystal Reports Fields/Functions in CRChart Macros

You can use any field or function in Crystal Reports as a parameter for a CRChart Macro.

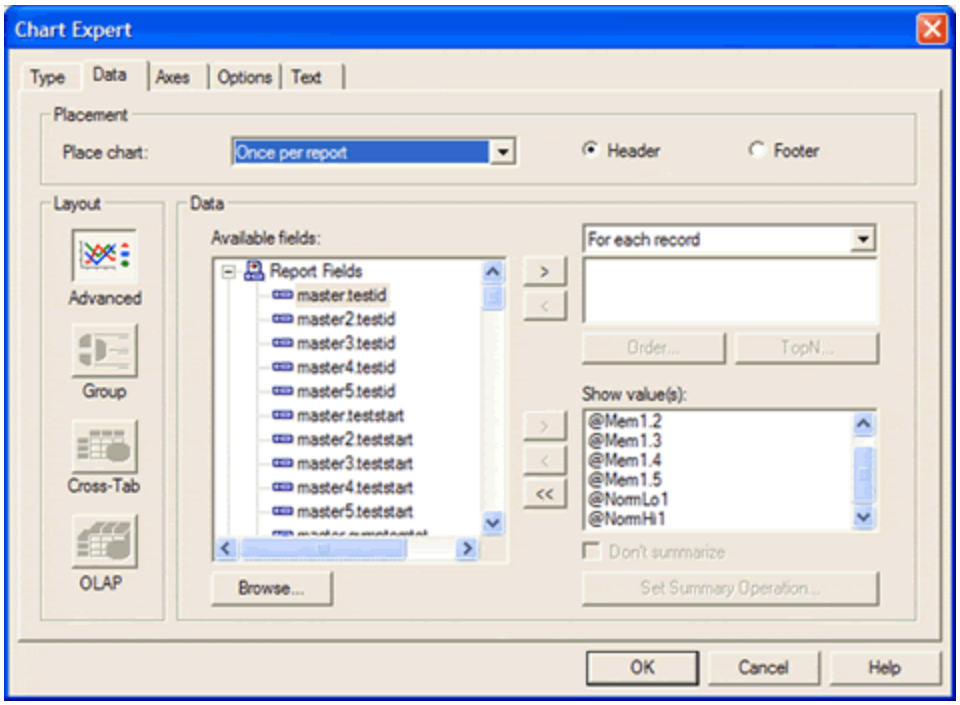
- Add the report field or function that you want to use with a CRChart macro to the end of "Show value(s)" list in the Data tab of the Chart Expert dialog.
- Make a note of the zero-based position of this field in the "Show value(s)" list.
- In your CRChart macro, use the letter "P" followed by the number that represents the position in the "Show value(s)" list.

IMPORTANT NOTES

- Do not reference an item in the "Show value(s)" list that is part of the dataset being graphed (i.e., do not use "P0"). This would cause that particular data series to disappear.
- In this version of CrChart, you cannot use the same "P" parameter twice (e.g., @USER_FILL P5 5 P5 5 is not valid). You could, however, drag the same field twice into the "Show value(s)" list so that they become P5 and P6 (e.g., @USER_FILL P5 5 P5 6 is valid).

Example:

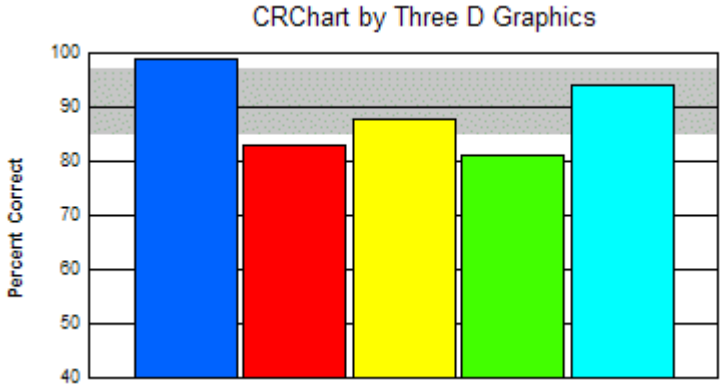
In the Data tab of the Chart Expert dialog, the "NormLo1" and "NormHi1" fields are added at zero-based positions 5 and 6 in the "Show value(s) list.



The following macro is entered in the Title field in the Text tab of the Chart Expert dialog.

```
@USER_FILL 0 1 P5 P6 192 192 192
```

This macro defines the location/size of a user-defined fill area. The first two parameters (0 and 1) define the start location on the chart frame. The next two parameters (P5 and P6) are taken from the values of the "NormLo1" and "NormHi1" fields in the "Show value(s)" list and define the ending location of the fill area. The last three parameters define the color of the fill area.



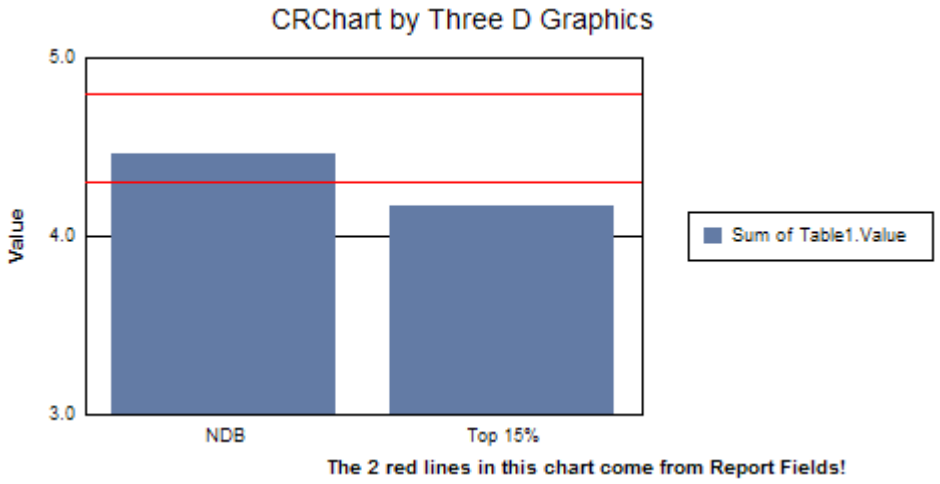
In the following example, the following fields are listed in the "Show value(s)" list in the Data Tab of Chart Expert:

```
Sum of Table1.Value - Position 0
Sum of Table1.ErrBarLo - Position 1
Sum of Table1.ErrBarHi - Position 2
```

In the Text Tab of Chart Expert, the following CRChart macro is defined in the Subtitle field:

```
@Y P2
@Y P1
```

These macros create horizontal user-defined lines at P2 and P1:



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 in multiple title fields. To avoid this potential conflict, the title fields are evaluated in the following order:

1. Y2 Axis Title
2. Y1 Axis Title
3. X Axis Title
4. Footnote
5. Subtitle
6. Title

For example, assume "@SWAP 0" is defined in the Footnote string and "@SWAP 1" is defined in the X Axis Title field. In this example, the "@SWAP 1" macro would be used because it is evaluated after the "@SWAP 0" 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 1".

Persistence

Many of the chart enhancement macros set a property of the chart and that property will remain set, even if the macro is removed from the field.

For example, if "@SWAP 1" is used to reverse the series/groups in the chart, the series/groups will remain reversed even if the "@SWAP 1" macro is removed. A "@SWAP 0" macro is needed to return the series/groups to their normal/default order.

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

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

Compatibility with Existing Chart Library

Your existing reports with charts in them will look like they always have. The Chart Expert and Chart User Interface in Crystal Reports work like they always have. If you don't use the special macro commands in the title fields, you will notice no difference in your charting functionality.

Error Reporting

If your macro encounters an error, a red 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 CRChart in Visual Basic

CRChart macros can be used in Chart Expert that comes with Crystal Reports. These fields can also be accessed programmatically via the ChartObject class. The following example code shows how CRChart macros can be used to change chart properties at runtime.

```
Dim crGraphObj As CRAXDRT.GraphObject
Set crGraphObj =
crChart.OpenSubreport.Sections(1).ReportObjects.Item(1)
crGraphObj.FootNote = "@BP @SZ 35 "
    ....
crGraphObj.FootNote = crGraphObj.FootNote & "@MS " & CStr(iRow
- Fractile135) & " " & CStr(iMarkerNumInSSCSDK80) & " "
    ....
'Set marker color in chart
crGraphObj.GroupsTitle = crGraphObj.GroupsTitle & "@MC " &
CStr(iRow - Fractile135) & " " & CStr(r) & " " & CStr(G) & " "
& CStr(B) & " "
    ....
'Control the Y1 axis scale on all plot areas
'Get Y Max and Y Min
GetCommonYMaxMin mvPlotYMaxMin, dYMax, dYMin
ScaleYMaxYMin dYMax, dYMin, strYMax, strYMin, iYDivision
    ....
crGraphObj.FootNote = crGraphObj.FootNote & "@SC " & strYMin &
", " & strYMax & " " & "@MK " &
CStr(UBound(vReportData_Table_Marker, 1) - Fractile135)
crGraphObj.DataAxisDivisionMethod = crManualDivision
crGraphObj.DataAxisDivisionNumber = iYDivision
```

Macro Definitions

@3DSCAT (3D Scatter Chart)

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

Syntax:

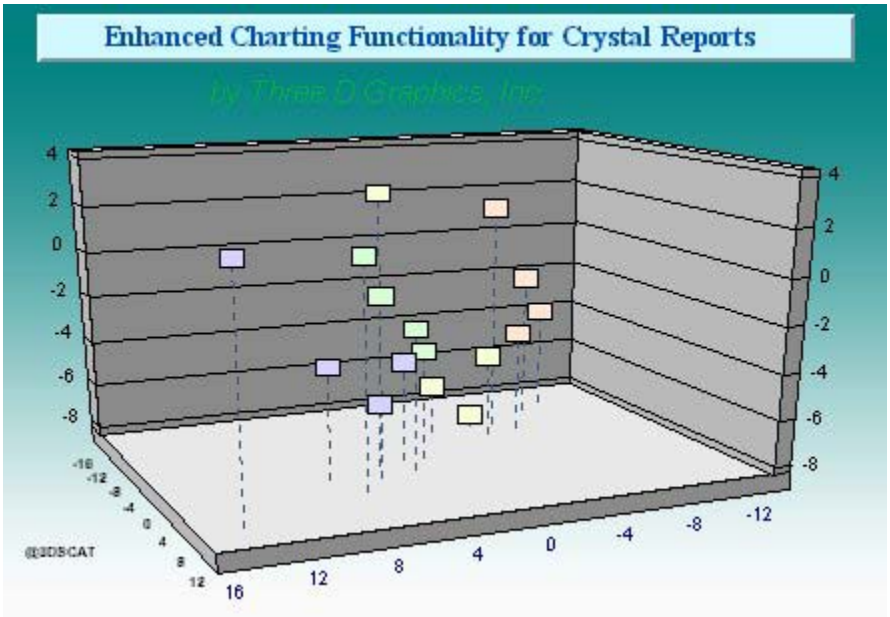
```
@3DSCAT
```

Parameters:

None

Example:

```
@3DSCAT
```



Persistent:

Yes

@AA (Auto Arrange)

This macro automatically arranges elements in a chart.

Syntax:

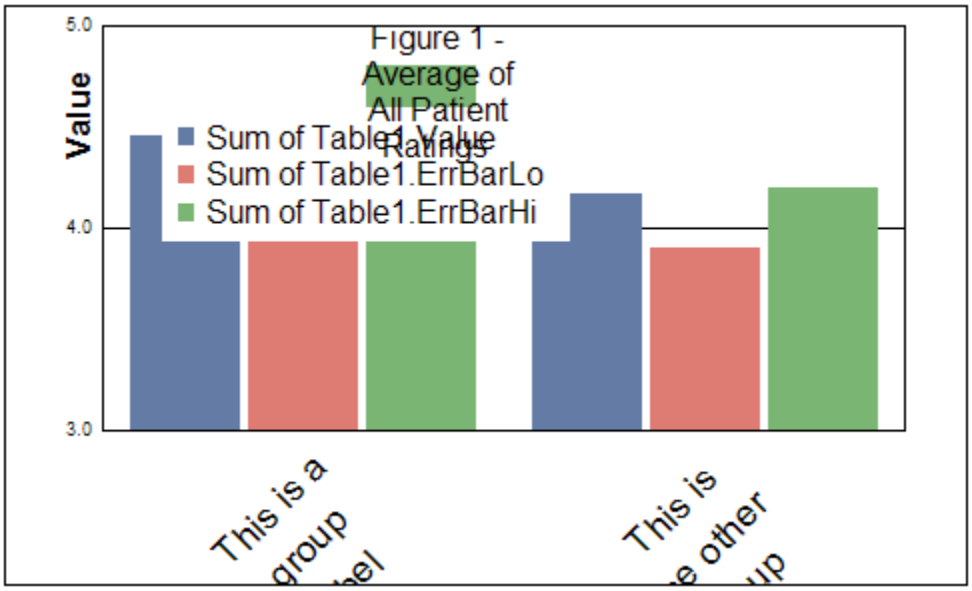
```
@AA
```

Parameters:

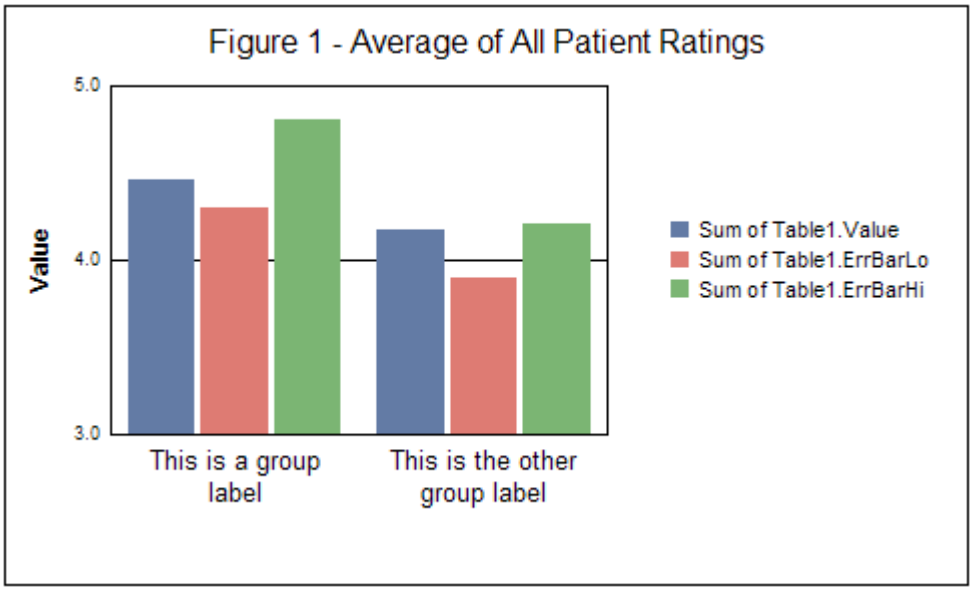
None

Example:

Before @AA



After @AA macro applied



Persistent:

Yes

@AGL (Alias Group Label)

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

Syntax:

```
@AGL n sz
```

Parameters:

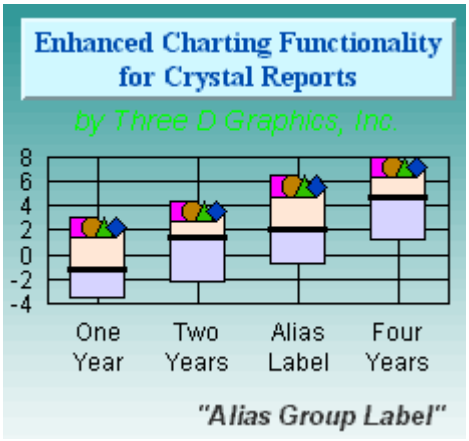
n; 0...number of groups in chart (0=Group 1). Defines the group on which to place the alias label.

sz; Optional group label. It must be terminated with a '~'

Example:

This example will create a box plot chart with T-Style 'tails' (@BP2), 4 markers (@MK 4), and an alias label on group number 2 with the text "Alias Label" (@AGL 2 Alias Label~).

```
@BP2
@MK 4
@AGL 2 Alias Label~
```



Persistent:

No

Also See:

@ASL

@ALPHA (Alpha Channel Transparency)

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

Syntax:

```
@ALPHA nSeries nValue
```

Parameters:

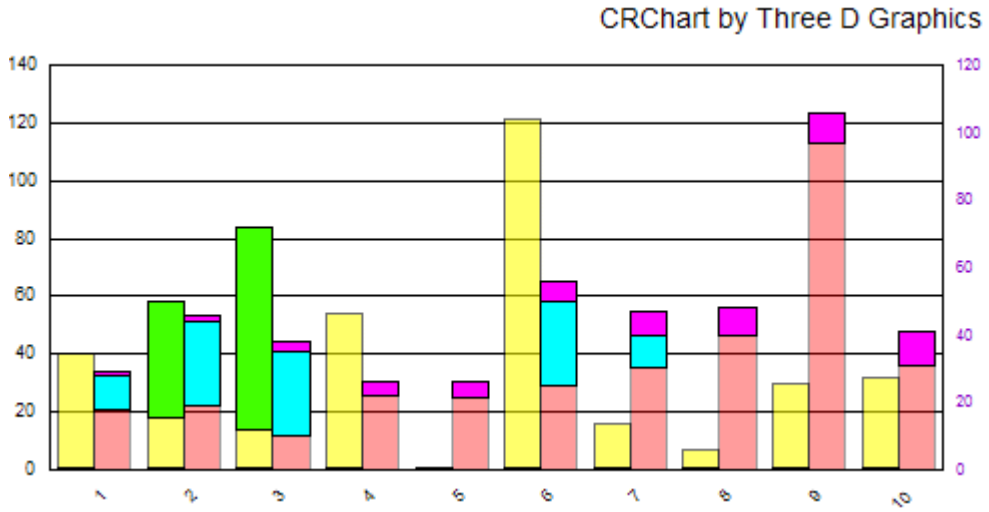
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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

Example:

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

```
@ALPHA 1 100
@ALPHA 2 150
```



Persistent:

Yes

@ASL (Alias Series Label)

This macro can be used to change a series label.

Syntax:

```
@ASL n sz
```

Parameters:

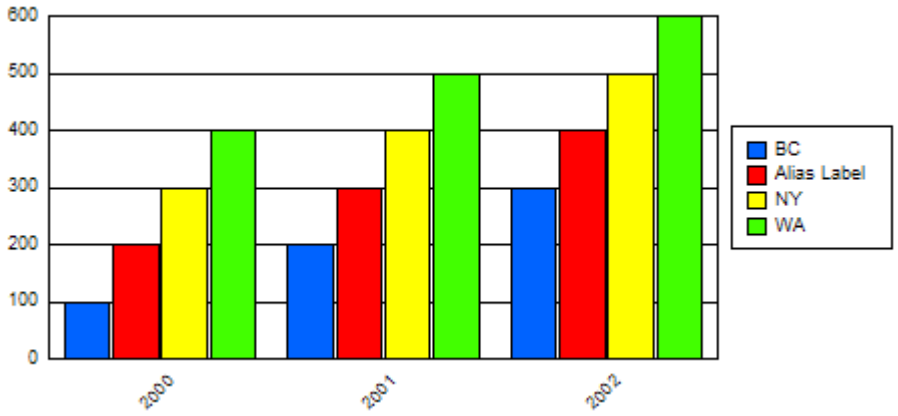
n; 0...number of series in chart (0=Series 1). Defines the series on which to place the alias label.

sz; Optional string. It must be terminated with a '~'

Example:

This example creates a chart with an alias label on series number 1 with the text "Alias Label".

```
@ASL 1 Alias Label~
```



Persistent:

No

Also See:

@AGL

@ASL_DP (Alias Series Label/Data Point)

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

Syntax:

```
@ASL_DP nS
```

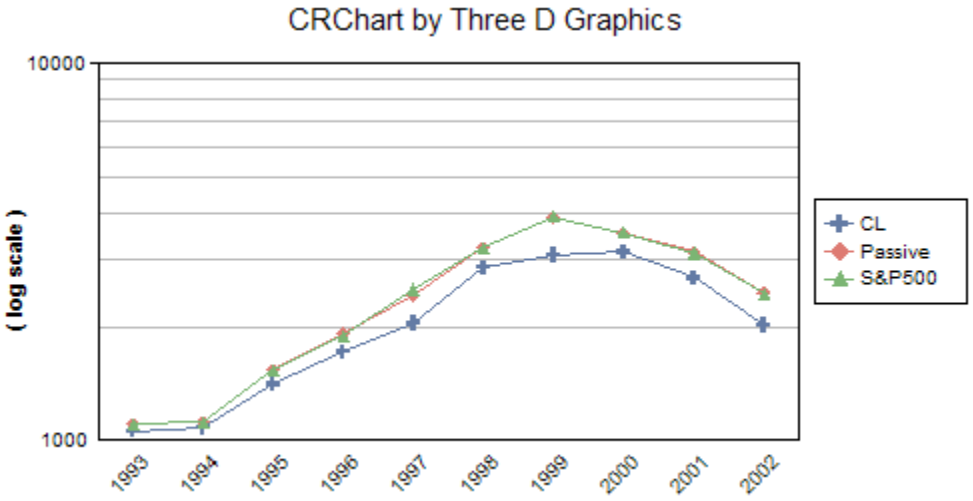
Parameters:

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

Example:

This example maps the sub string from group label zero to the first element in the legend:

```
@ASL_DP 0
@ASL 1 Passive~
@ASL 2 S&P500~
@SC 1000 5000
```



Persistent:

No

Also See:

@ASL

@AXIS (Assign Series to Axis)

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

Syntax:

```
@AXIS nSeries bY2
```

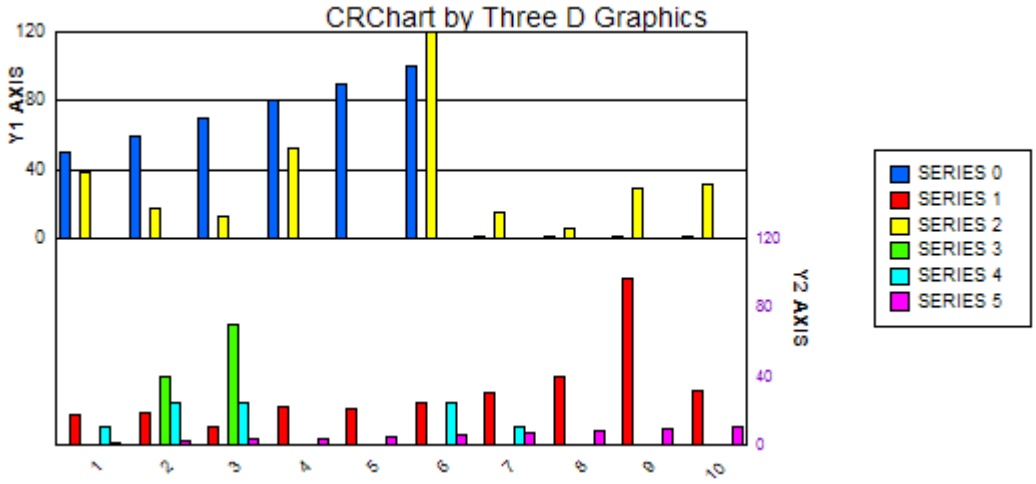
Parameters:

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.

bY2; 0=assign *nSeries* to Y1-Axis, 1=assign *nSeries* to Y2-Axis

Example:

```
@GRAPHTYPE 18
@AXIS 0 0
@AXIS 1 1
```



Persistent:

Yes

@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
```

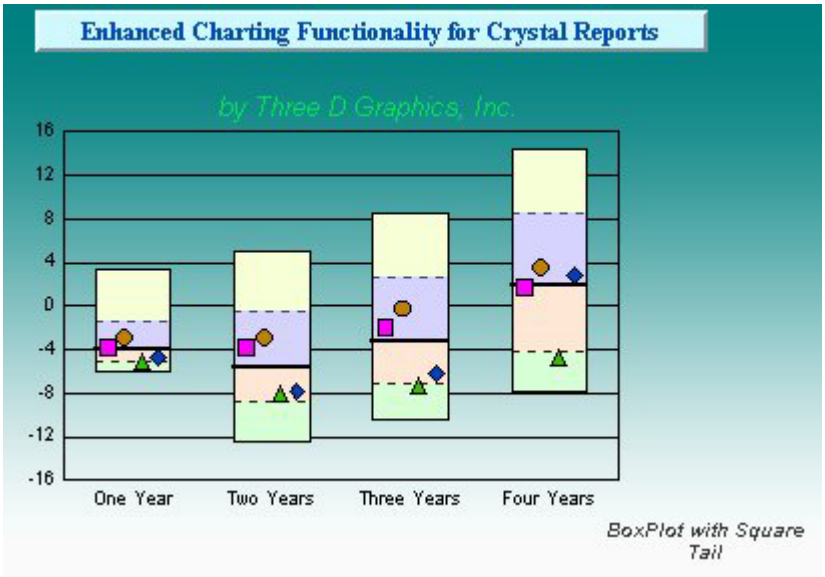
Parameters:

None

Example:

This example creates a standard box plot (@BP) with 4 markers (@MK 4) on top of the boxes.

```
@BP
@MK 4
```



Persistent:

No

Also See:

@BP2, @BP3

Notes:

The default marker shape for markers on box plots are: Series 1 – Rectangle, Series 2 – Circle, Series 3 - Triangle "tip up", Series 4 – Diamond, Series 5 – Five Pointed Star, Series 6 – Thin Plus Sign, Series 7 – "X-Shape", and Series 8 – Triangle "tip down". These default marker shapes can be changed using the "@MS" macro.

@BP2 (Box Plot with T-Style Tail)

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

Syntax:

```
@BP2
```

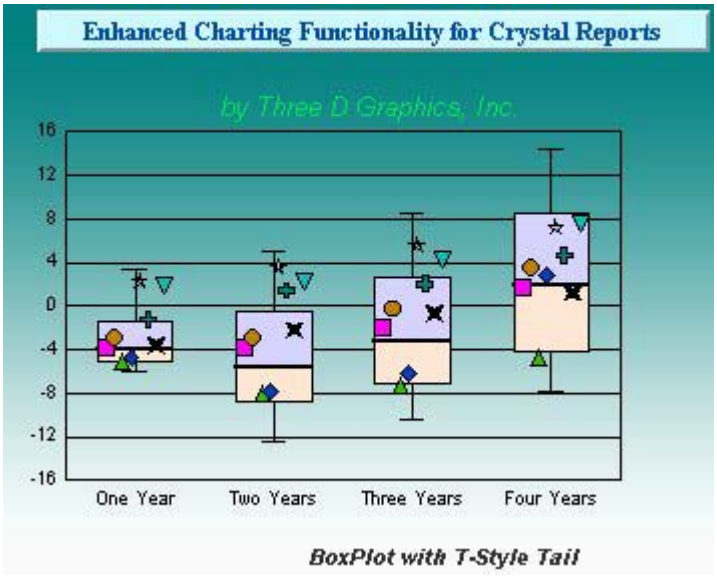
Parameters:

None

Example:

@BP2 creates a box plot with T-Style 'tails'. @MK8 creates eight markers on top of each box plot.

```
@BP2
@MK 8
```



Persistent:

No

Also See:

@BP, @BP1, @BP3

Notes:

The default marker shape for markers on box plots are: Series 1=Rectangle, Series 2=Circle, Series 3=Triangle "tip up", Series 4=Diamond, Series 5=Five Pointed Star, Series 6=Thin Plus Sign, Series 7=X-Shape, and Series 8=Triangle "tip down". These default marker shapes can be changed using the "@MS" macro.

@BP3 (Box Plot with I-Style Tail)

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

Syntax:

```
@BP3
```

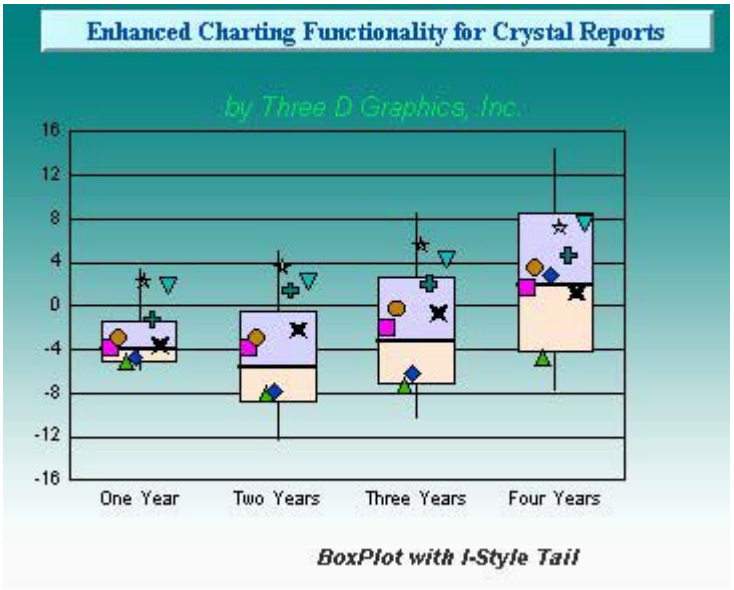
Parameters:

None

Example:

In this example, @BP3 creates a box plot with an I-Style tail. @MK8 creates eight markers on top of each box plot.

```
@BP3
@MK8
```



Persistent:

No

Also See:

@BP, @BP1, @BP2

Notes:

The default marker shape for markers on box plots are: Series 1=Rectangle, Series 2=Circle, Series 3=Triangle "tip up", Series 4=Diamond, Series 5=Five Pointed Star, Series 6=Thin Plus Sign, Series 7=X-Shape, and Series 8=Triangle "tip down". These default marker shapes can be changed using the "@MS" macro.

@COLOR_MODE (Color Mode)

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

Syntax:

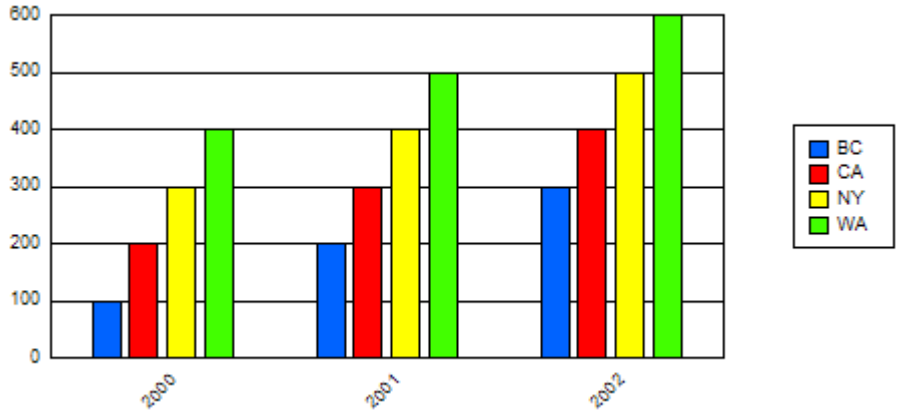
```
@COLOR_MODE nMode
```

Parameters:

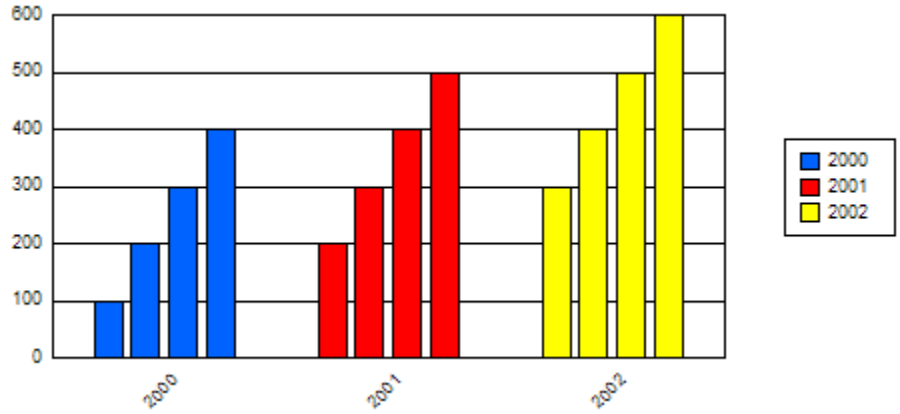
nMode; 0 = "Color By Series", 1= "Color By Group"

Example:

```
@COLOR_MODE 0
```




```
@COLOR_MODE 1
```



Persistent:

No

@COMBO (Combo Chart)

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

Syntax:

```
@COMBO nSeries nStyle
```

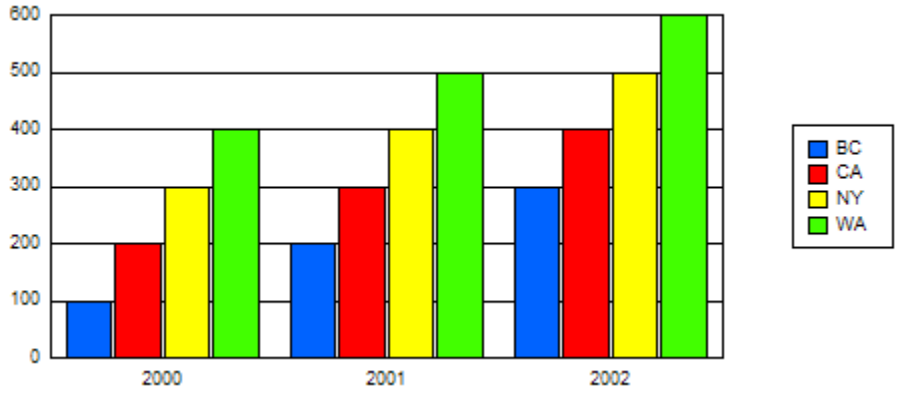
Parameters:

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.

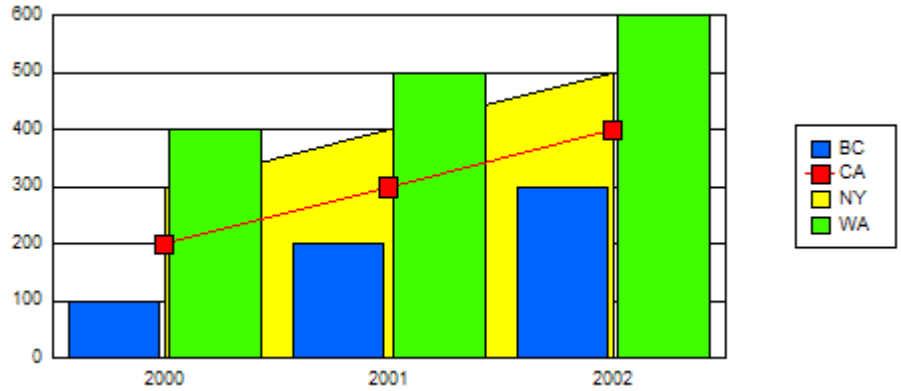
nStyle; 0..2 (0= Bar, 1=Line, 2=Area)

Example:

```
@COMBO -1 0
```




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



Persistent:

Yes

@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.

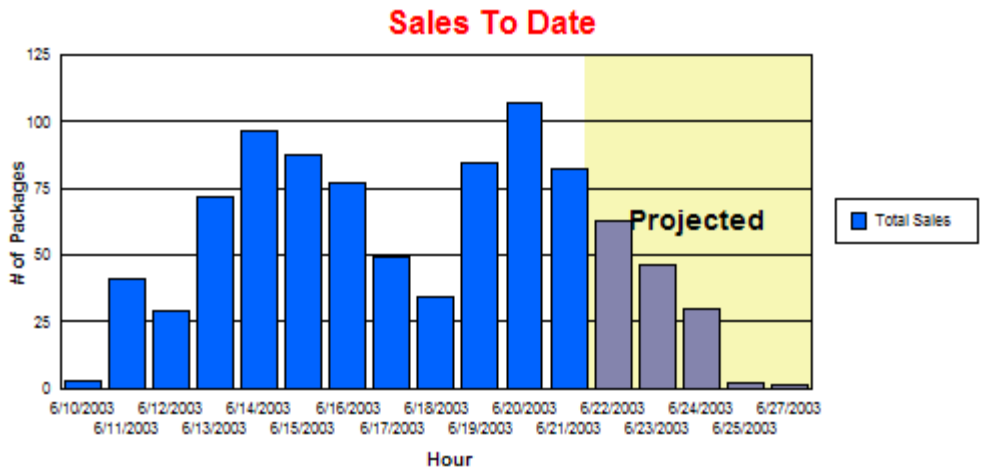
7 = force all series in *nGroup* to the conditional color regardless of *fValue*.

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

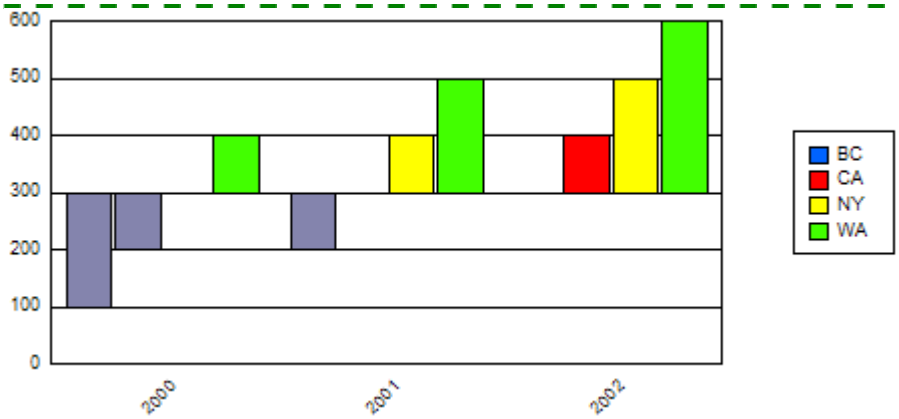
nRed, nGreen, nBlue; 0...255 specifies the RGB value of the Conditional color.

Examples:

```
@USER_FILL .7 1 0 1 245 245 180 Projected
@COND_COLOR -1 12 6 0 133 133 173
@ASL 0 Total Sales
```




```
@Y1BASE 300
@COND_COLOR -1 -1 0 300 133 133 173
```



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*).

@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:

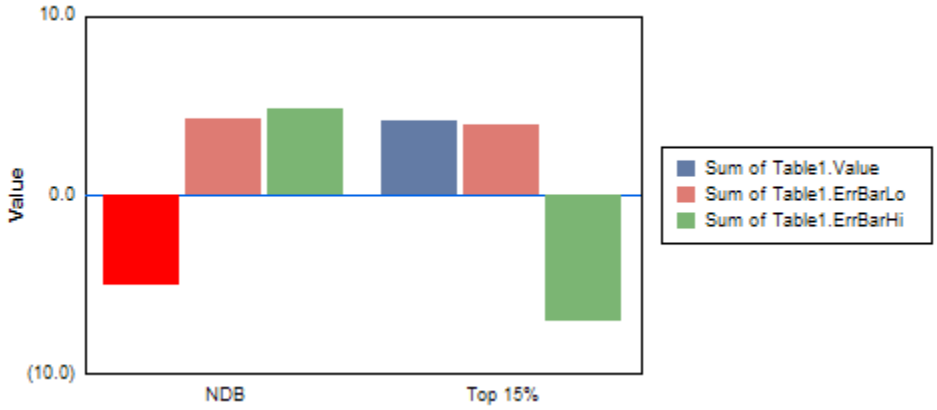
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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

szGroupLabel; Group label string

Example:

```
@COND_GROUP_LABEL 0 255 0 0 NDB
```



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.

@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 use must prefix the target label and a tilda (~) to each group label. When the macro finds a group label that matches the prefix, the color is applied to the riser.

Syntax:

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

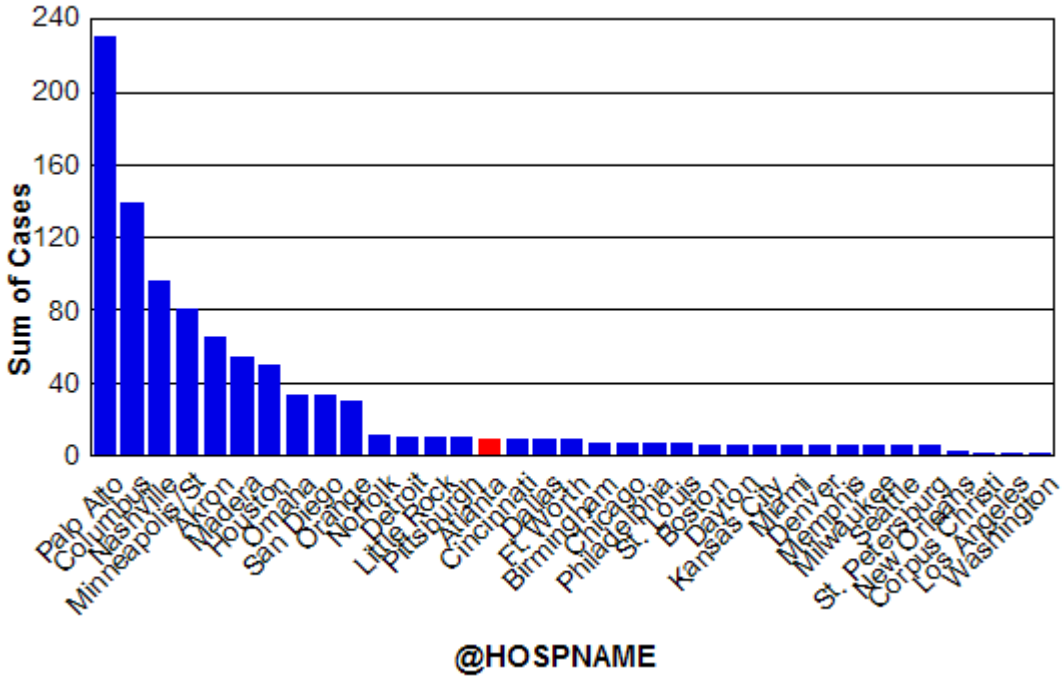
Parameters:

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

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

Example:

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



In this example, the "Atlanta~" string is prefixed to every group label in this chart.

Persistent:

No

@CXY (Color X/Y User Defined Line)

This macro is identical to @XY, except it allows you to specify the RGB color of the line to be created.

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

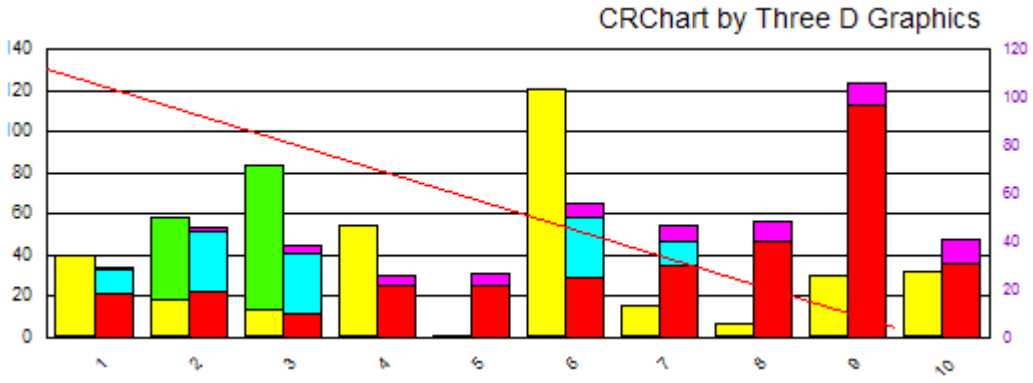
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

Example:

```
@CXY 0 130 0.9 4.5 255 0 0
```



Persistent:

No

@CY (Color Y User Defined Line)

This macro is identical to @Y except it allows you to specify the RGB color of the line to be created.

Syntax:

```
@CY n nRed nGreen nBlue
```

Parameters:

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

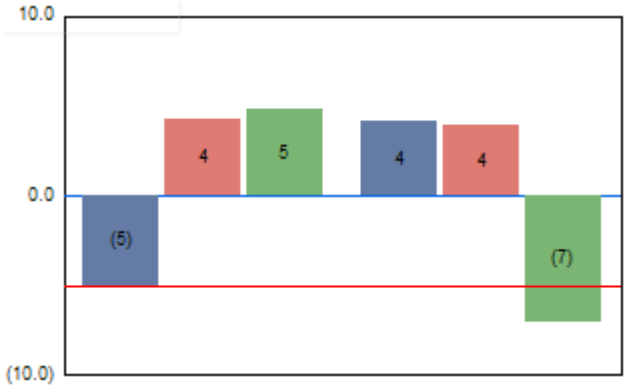
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

Syntax:

```
@CY -5 255 0 0
@NEG_STYLE 1
```



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
```

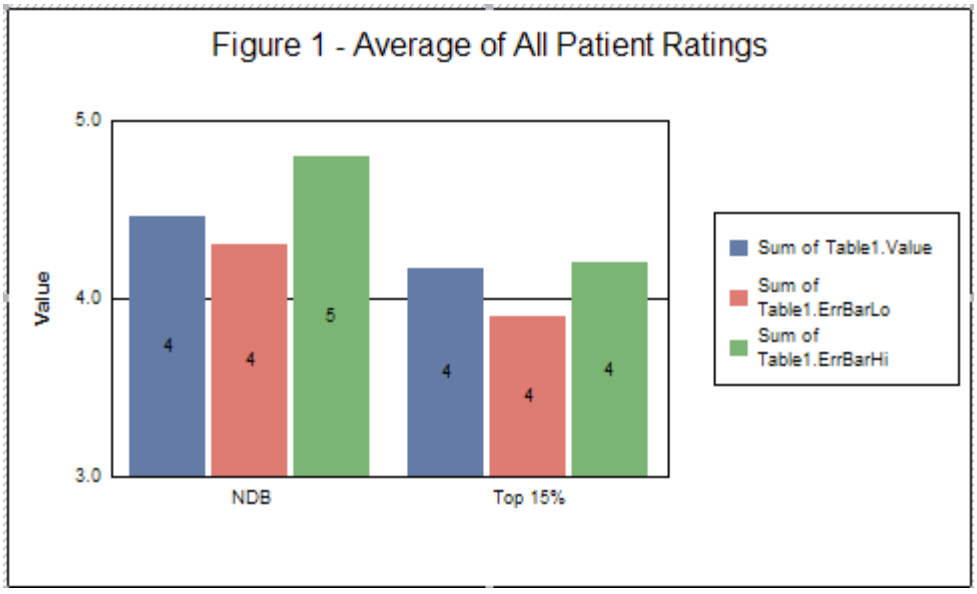
Parameters:

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	Show value(s) of data item(s).
2	Show unique data item text.
3	Show values and non-numeric text
4	Show absolute value in stacked segments.
5	Show unique data item text and show absolute value in stacked segments
7	Show values and non-numeric text and show absolute value in stacked segments
8	Total value on top of the stacked bar.
16	Display the Z-value in bubble charts.
32	Calculate and show the riser value as a percentage
64	Display the 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 type
```

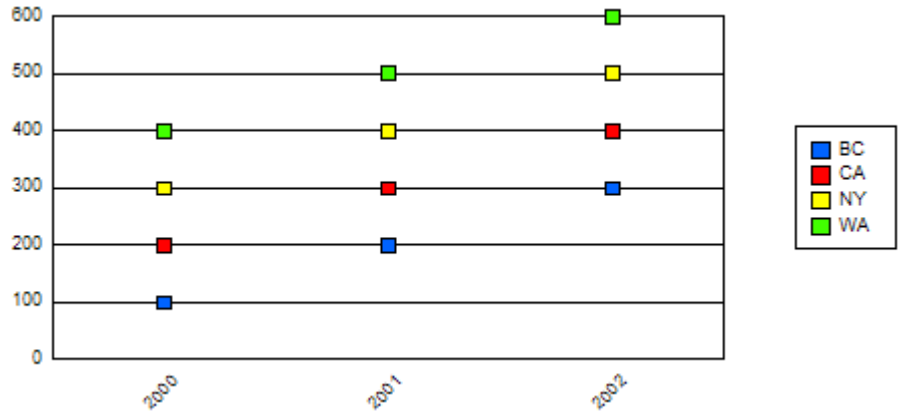
Parameters:

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.

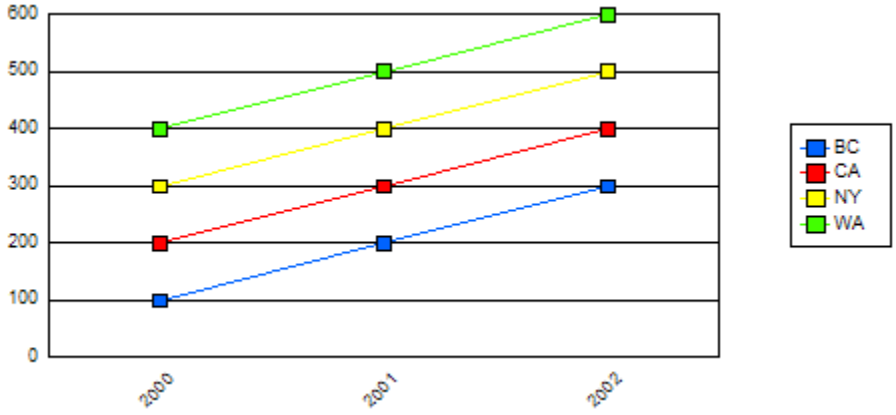
type; 1 = draw markers only, 2 = draw lines only, 3 = draw markers and lines.

Example:

```
@DLT -1 1
```




```
@DLT -1 3
```



Notes:

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

Persistent:

Yes

@DEBUG (Show Debug Information)

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

Syntax:

```
@DEBUG
```

Parameters:

None

Persistent:

No

@DP (Data Point)

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:

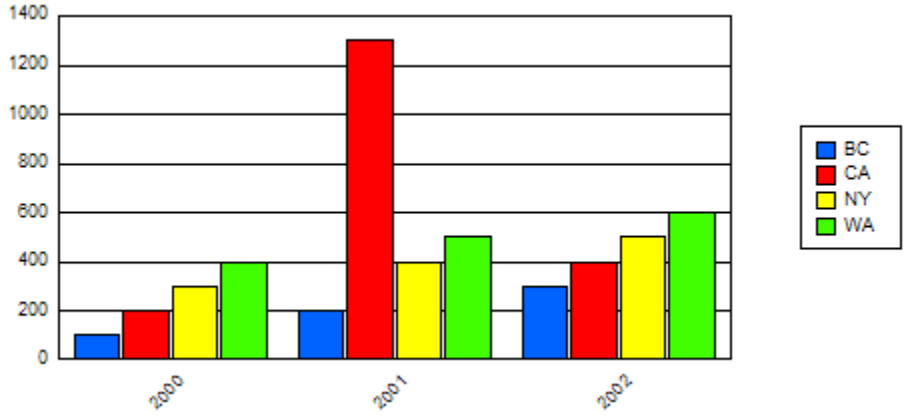
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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 1 1 1300
```



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
```

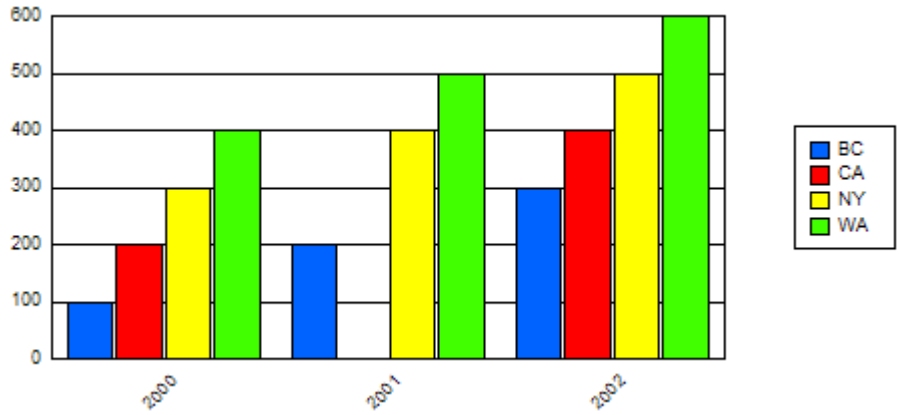
Parameters:

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:

```
@DPC 1 1
```



Persistent:

No

Also See:

@DP

@DX (Divisions on X-Axis)

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

Syntax:

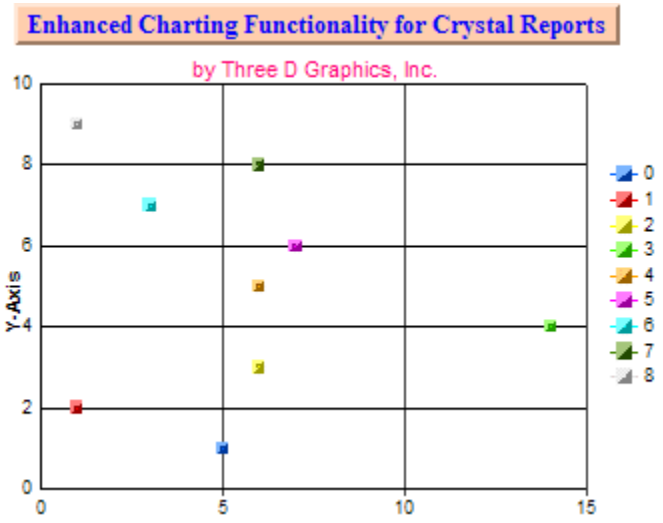
```
@DX n
```

Parameters:

n; Number of divisions on the X-Axis

Example:

```
@DX 2
```



Persistent:

Yes

Also See:

@DY

@DY (Divisions on Y-Axis)

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

Syntax:

```
@DY n
```

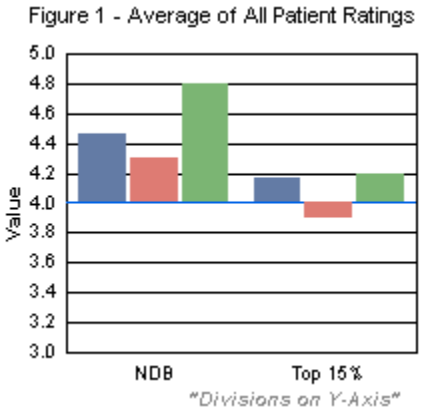
Parameters:

n; Number of divisions on the Y-Axis

Example:

This example sets the Y1 baseline at 4.0 (@Y1Base 4.0) and the number of divisions on the Y-Axis to nine (@DY 9).

```
@Y1BASE 4.0
@DY 9
```



Persistent:

Yes

Also See:

@DX

@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
```

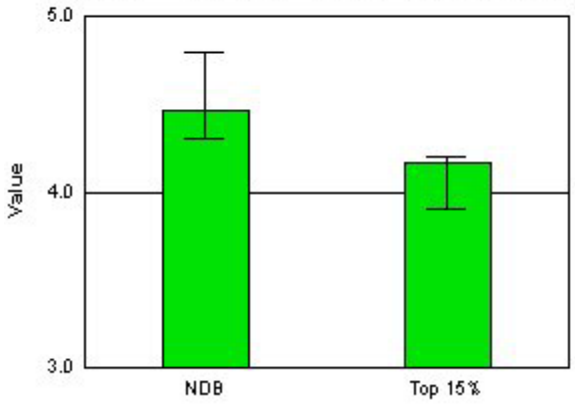
Parameters:

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

Example:

```
@EB 1
@HAT 68
```

Figure 1 - Average of All Patient Ratings



Persistent:

No

Also See:

@HAT (Hat on Error Bars)

@FORECAST (Add Blank Groups)

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

Syntax:

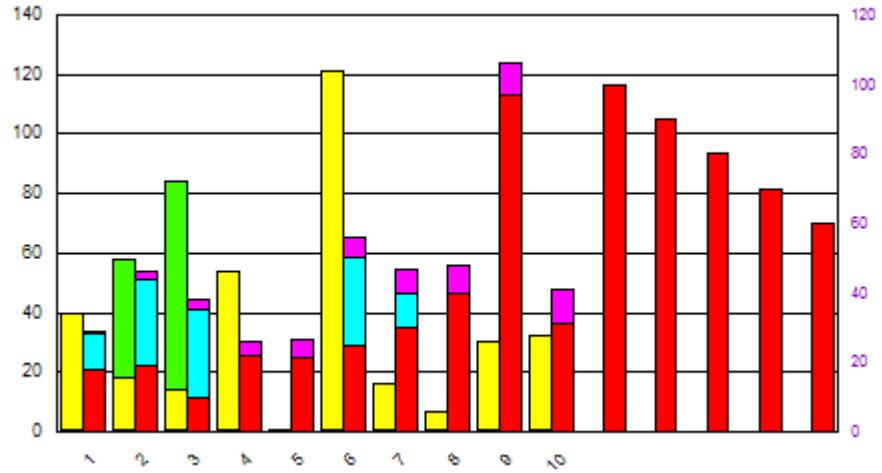
```
@FORECAST nPeriods
```

Parameters:

nPeriods; 0…1024 groups

Example:

```
@FORECAST 5
@DP 1 10 100
@DP 1 11 90
@DP 1 12 80
@DP 1 13 70
@DP 1 14 60
```



Persistent:

No

Also See:

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

@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:

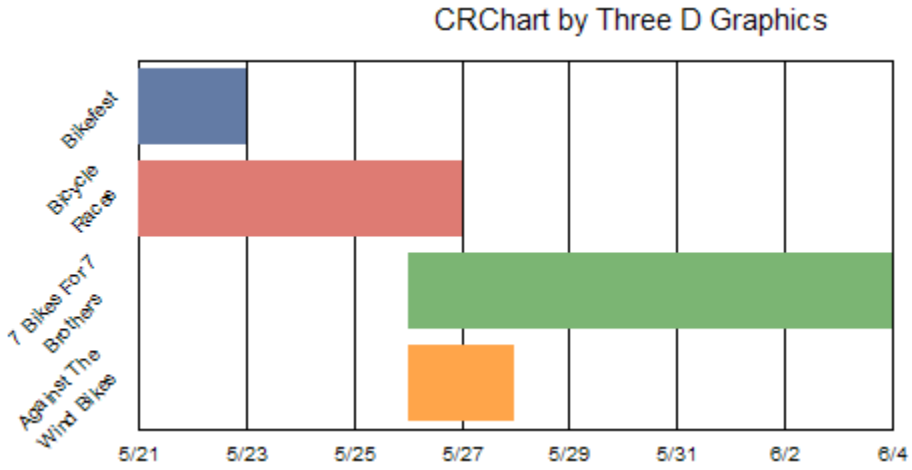
bTimeAxis; 1 = use time axis, 0 = do not use time axis

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

Syntax:

```
@GANTT 1 1 1
```



Persistent:

Yes

@GCOLOR (Chart Object Color)

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

Syntax:

```
@GCOLOR nObject nR nG nB
```

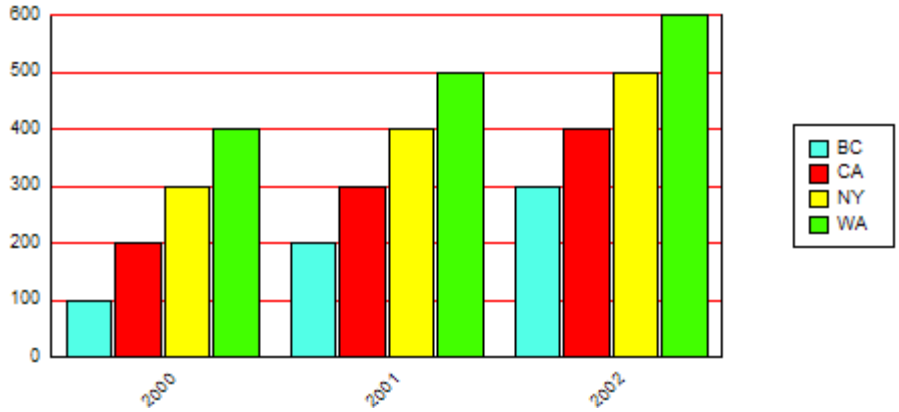
Parameters:

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

nR: 0...255 defines the Red portion of RGB color selection.
nG: 0...255 defines the Green portion of RGB color selection.
nB: 0...255 defines the Blue portion of RGB color selection.

Example:

```
@GCOLOR 12 255 0 0
```



Persistent:

Yes

@GM (Get Matrix)

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

Syntax:

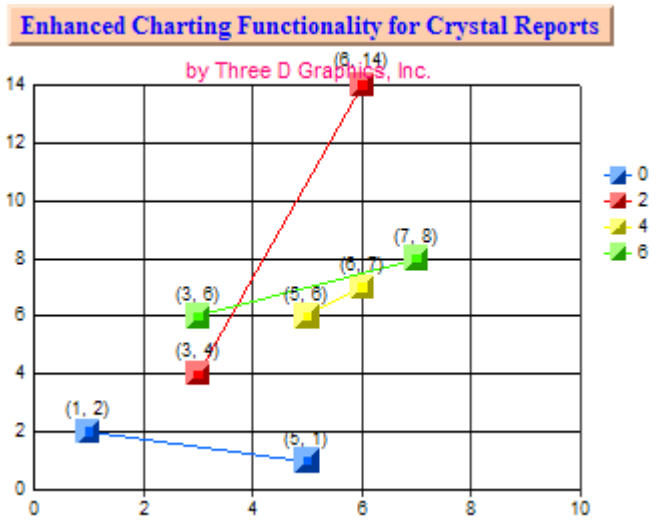
```
@GM n
```

Parameters:

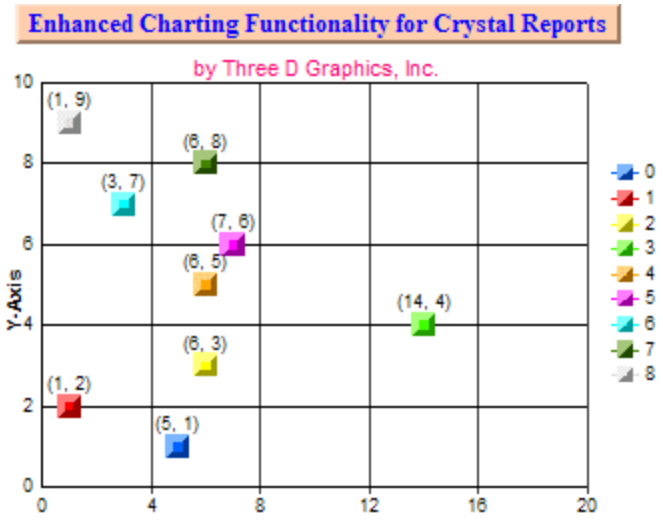
n; 1 = true (read data in column major order), 0 = false (read data in row major order)

Example:

```
@GM 0
```




```
@GM 1
```



Persistent:

No

@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

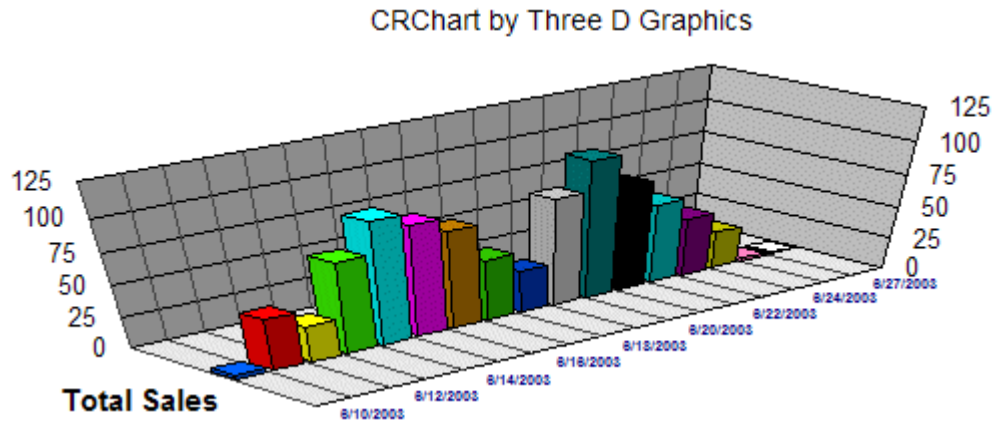
Value	Chart
0	Vertical Area Absolute
1	Vertical Area Stacked
2	Vertical Area Bi-Polar Absolute
3	Vertical Area Bi-Polar Stacked
4	Vertical Area Dual-Y Absolute
5	Vertical Area Dual-Y Stacked
6	Vertical Area Percent
7	Horizontal Area Absolute
8	Horizontal Area Stacked
9	Horizontal Area Bi-Polar Absolute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

Value	Chart
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
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

Value	Chart
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 73
```



Persistent:

Yes

@GX (Grid X-Axis Style)

This macro sets the Grid/Tick style on the X-Axis. It can only be used in a chart with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.).

Syntax:

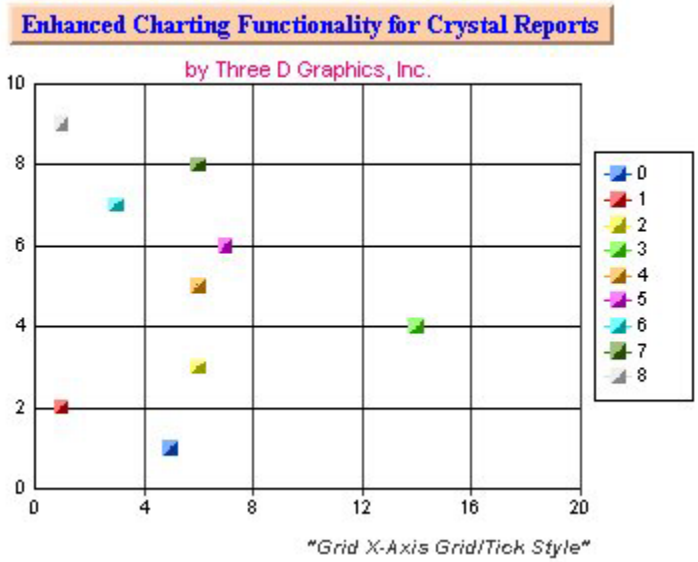
```
@GX n
```

Parameters:

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

Example:

```
@GX 2
```



Persistent:

No

Also See:

@GY

@GY (Grid Y-Axis Style)

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

Syntax:

```
@GY n
```

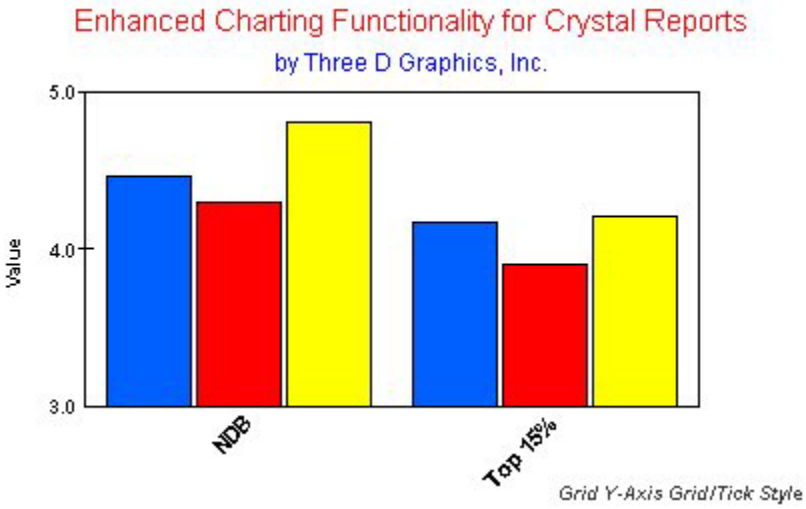
Parameters:

n; 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:

This example creates a chart with the grid/tick style 5; no grid lines, inner and outer ticks on the Y-Axis (the left side of this example chart).

```
@GY 5
```



Persistent:

No

Also See:

@GX

@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:

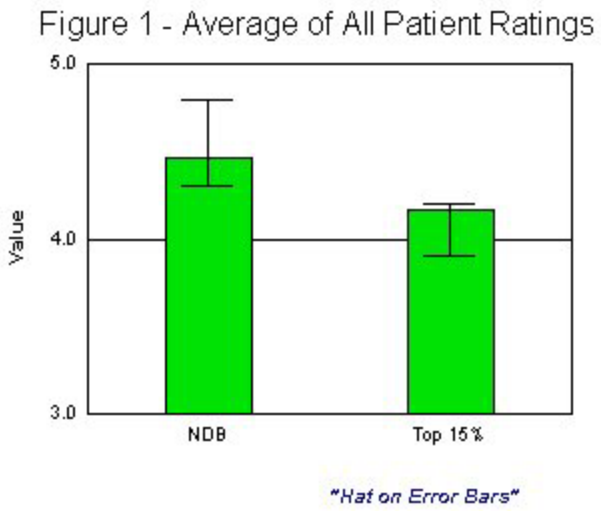
```
@HAT n
```

Parameters:

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

Example:

```
@EB 1
@HAT 90
```



"Hat on Error Bars"

Persistent:

No

Also See:

@EB

@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:

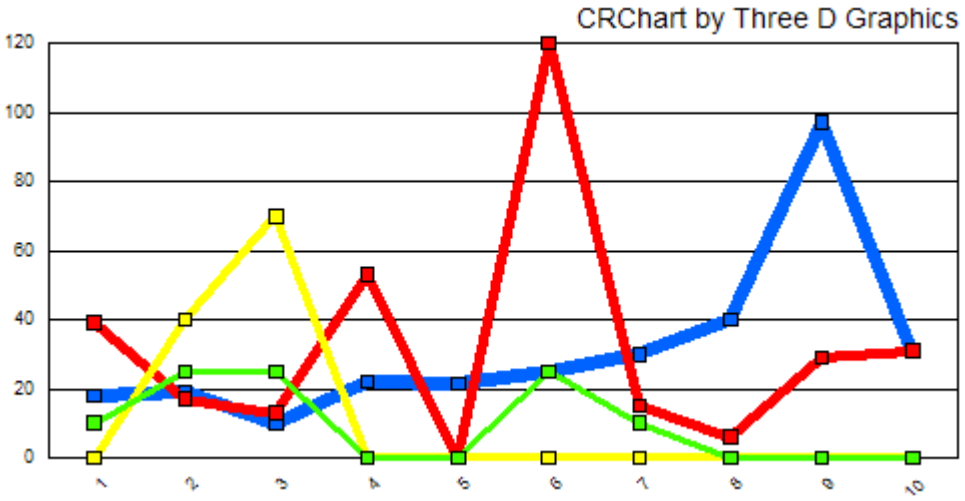
```
@HIDE_ZERO
```

Parameters:

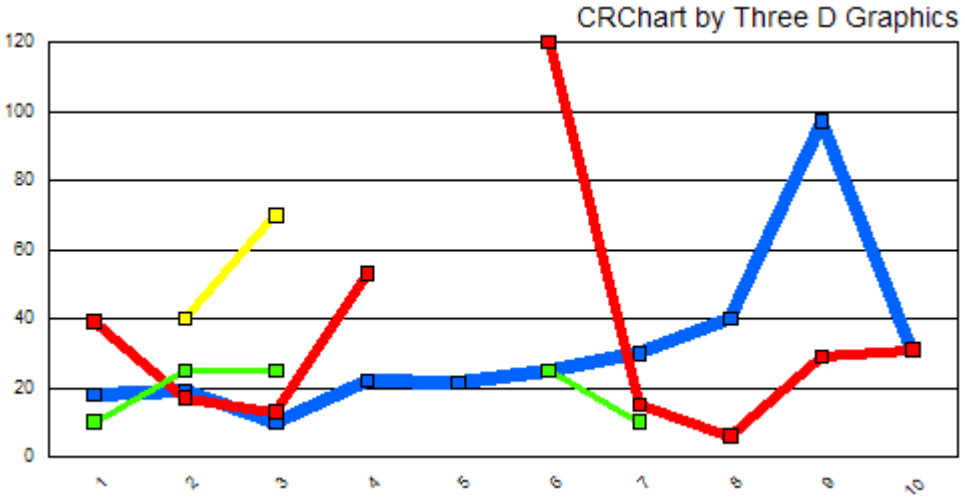
None

Example:

Before @HIDE_ZERO



After @HIDE_ZERO



Persistent:

No

@HL (Highlight Riser/Marker)

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

Syntax:

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

Parameters:

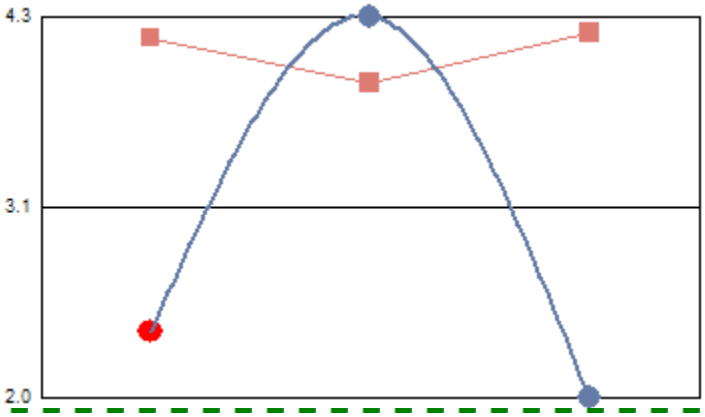
nSeries; Series #

nGroup; Series #

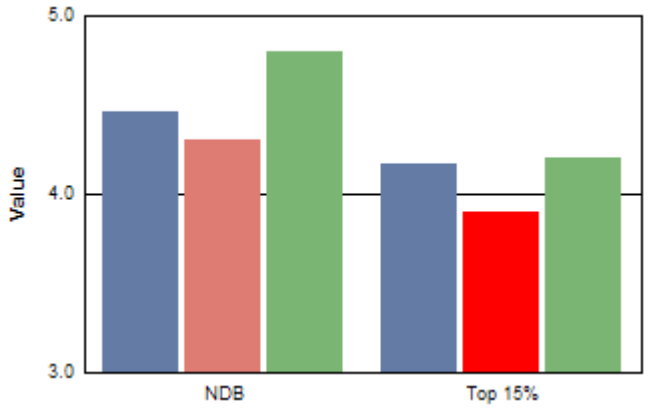
nRed, nGreen, nBlue: 0...255 specifies the RGB value to be applied to the riser/marker at *nSeries* and *nGroup*.

Example:

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




```
@HL 1 1 255 0 0
```



Persistent:

Yes

@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
```

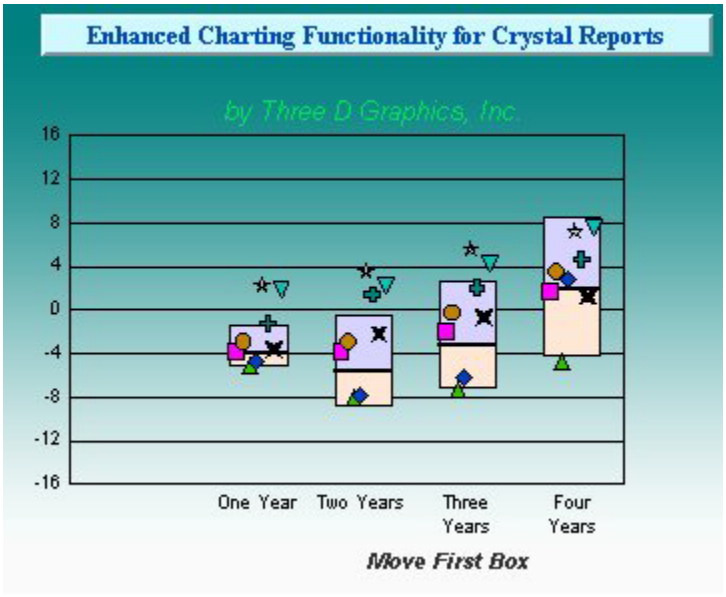
Parameters:

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

Example:

This example creates a box plot chart with the first box moved to the right by 5000 virtual units.

```
@BP
@IN 5000
```



Persistent:

No

Also See:

@BP, @BP1, @BP2, @BP3

@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.

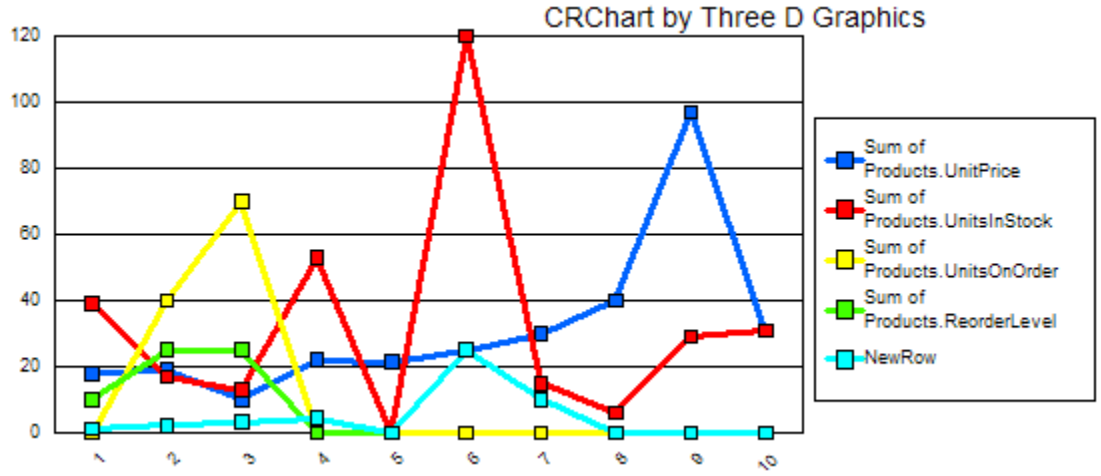
nElements; 1...1024 defines the number of fValues that follow. For example if *nElements* is 3, it must be followed by 3 *fValues* that will be assigned to the first 3 groups of the new series.

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

szSeriesName; Name of the new series what will appear in the legend.

Example:

```
@IR -1 4 1.1 2.2 3.3 4.4 NewRow~
```



Persistent:

Yes

@IS (Ignore Series)

This macro sets the specified series *n* 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:

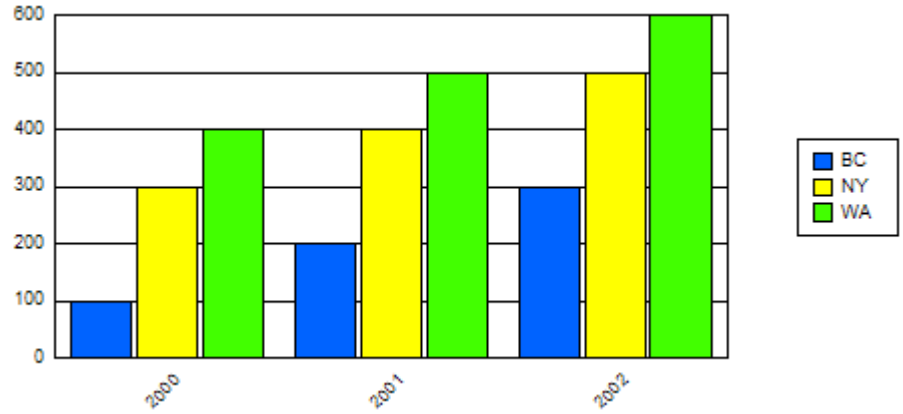
```
@IS n
```

Parameters:

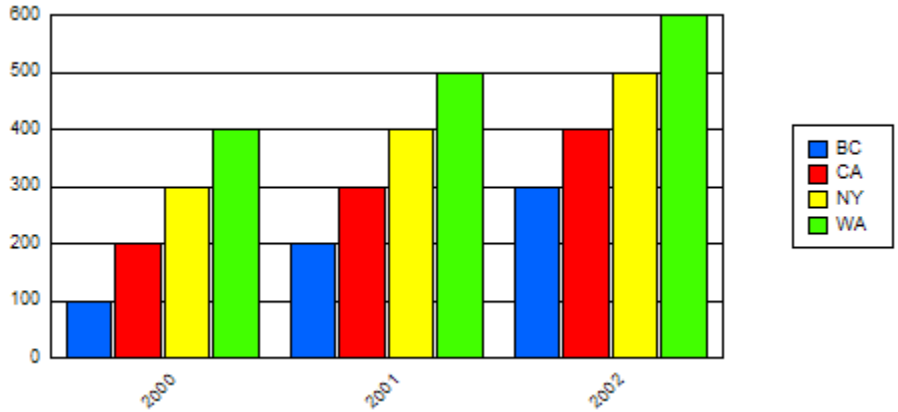
n; -1...*n* (where: *n* = the total number of series in the chart). -1=restore all series that were previously ignored, 0=Series 1, 1=Series 2, etc.

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:

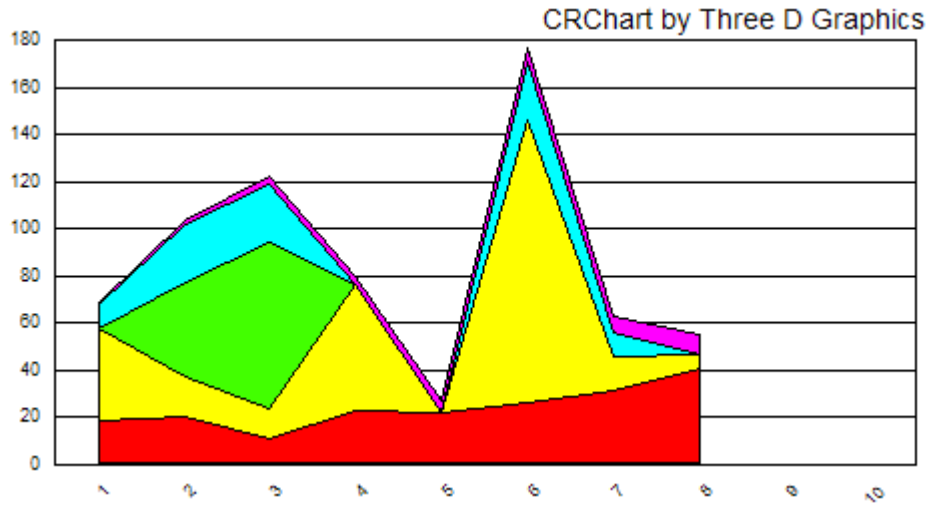
```
@LIMIT_VISIBLE_GROUPS  nGroup
```

Parameters:

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

Example:

```
@GRAPHTYPE 1
@LIMIT_VISIBLE_GROUPS 8
```



Notes:

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

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

Persistent:

Yes

@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.

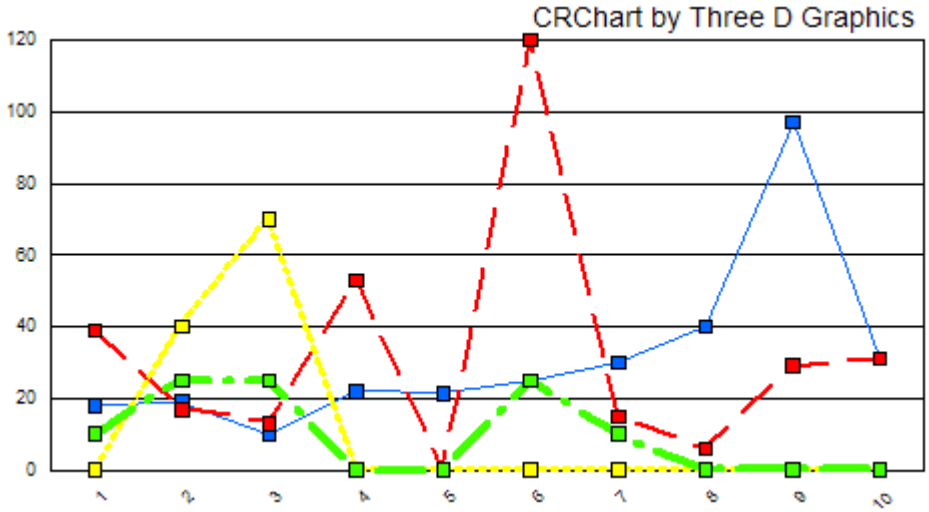
nThickness; 0...1000 selects the thickness of the line.

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 0 100 0 @LS 1 200 1 @LS 2 300 2 @LS 3 400 3 @LS 4 500 4
```



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.

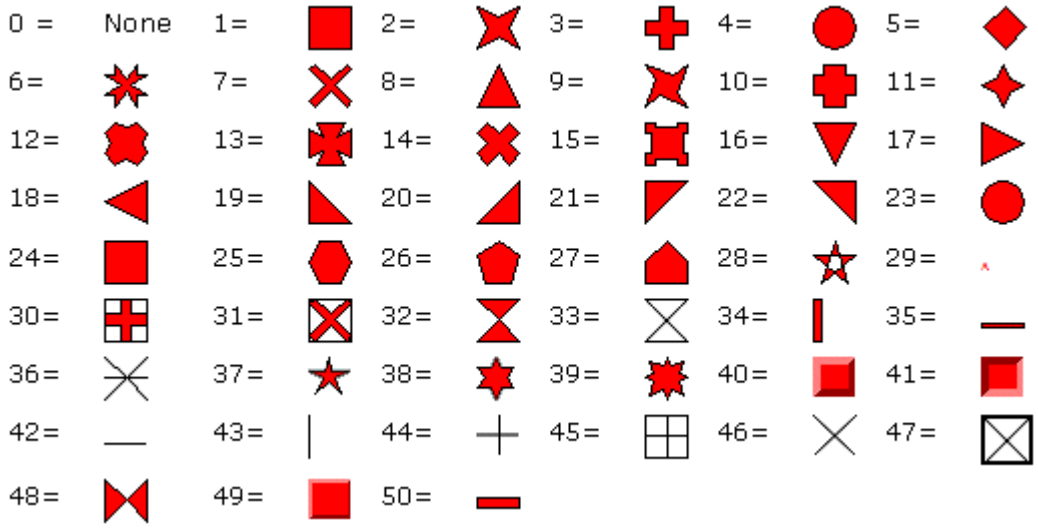
Syntax:

@MARKER *nSeries nMarker*

Parameters:

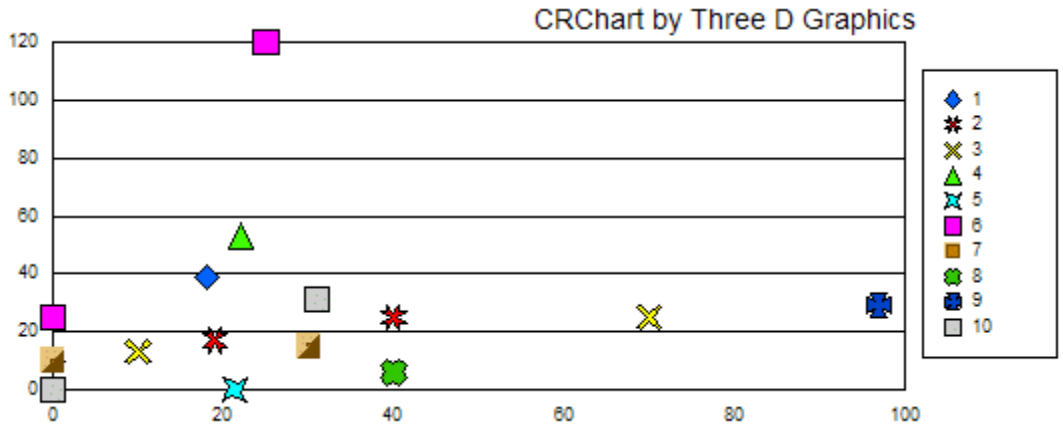
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nMarker; 0...50. Selects one of the following markers to apply to *nSeries*.



Example:

```
@GRAPHTYPE 50 @SZ 50 @MARKER 0 5 @MARKER 1 6 @MARKER 2 7
@MARKER 3 8 @MARKER 4 9 @ MARKER 5 31
@MARKER 6 40 @MARKER 7 12 @MARKER 8 13
```



Persistent:

Yes

@MC (Marker Colors for Box Plots)

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

Syntax:

```
@MC n nR nG nB
```

Parameters:

n; Series Number (1...8)

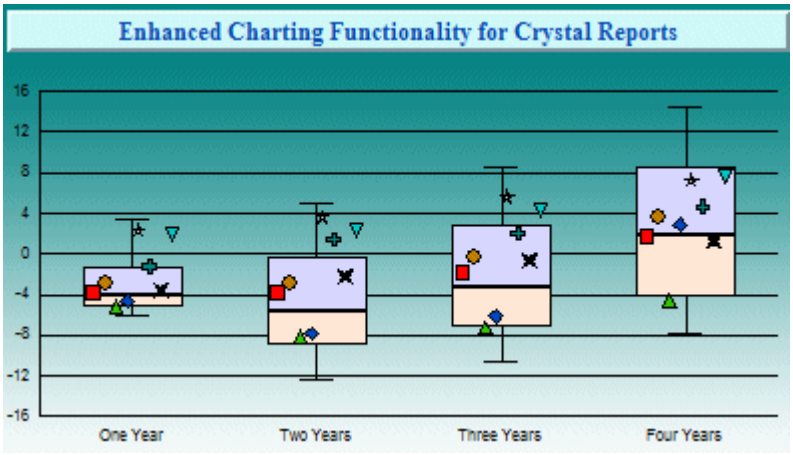
nR; 0...255 defines the Red portion of RGB color selection.

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

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

Example:

```
@BP2 @SZ 24 @MC 1 255 0 0
```



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.

@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 nR nG Nb
```

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.

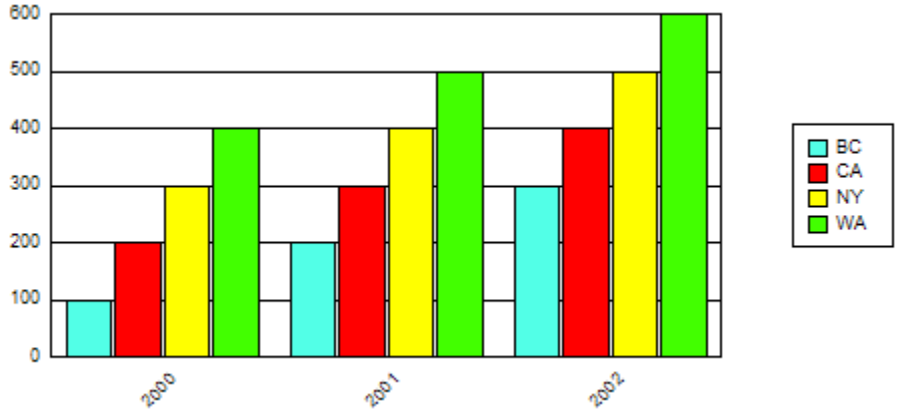
nR; 0...255 defines the Red portion of RGB color selection.

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

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

Example:

```
@MCOLOR 0 87 251 226
```



Persistent:

Yes

Also See:

@GCOLOR to change the color of other chart objects.

@MEAN (Mean Line)

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

Syntax:

```
@MEAN nSeries bShow
```

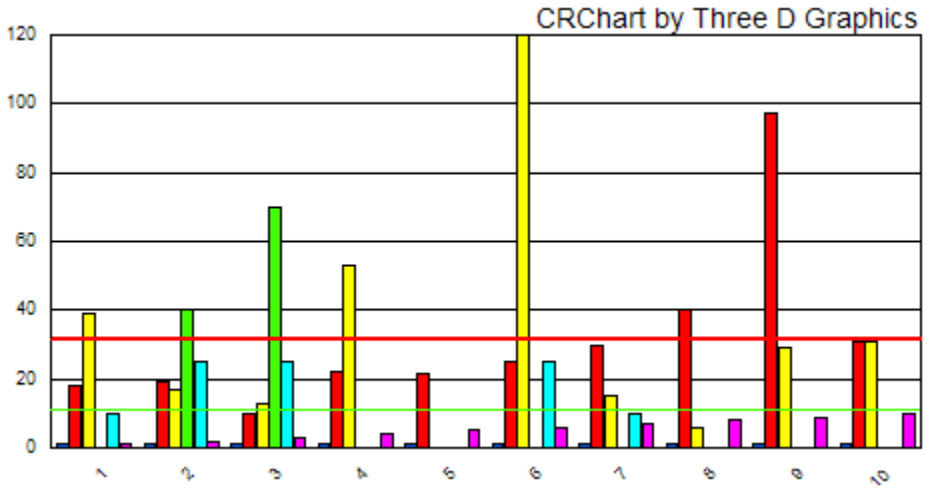
Parameters:

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

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

Example:

```
@MEAN 1 1
@MEAN 3 1
```



Persistent:

Yes

@MK (Number of Markers)

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

Syntax:

```
@MK n
```

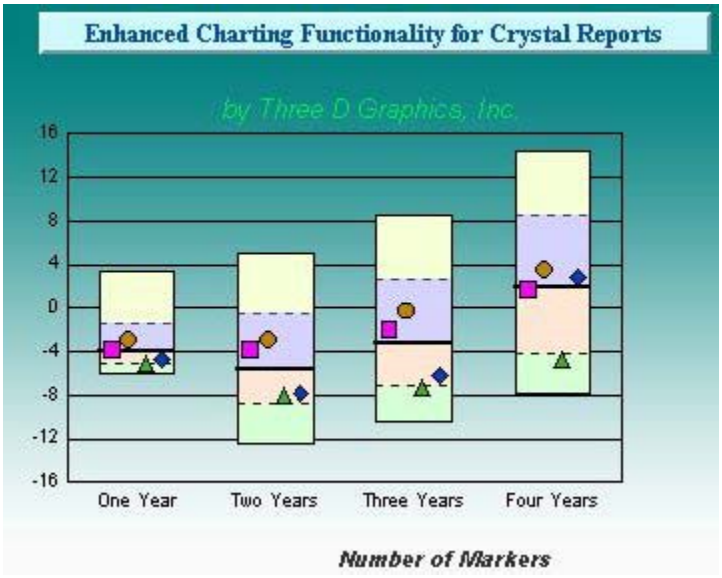
Parameters:

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

Example:

This example will create a standard box plot (@BP) with 4 markers (@MK 4) on top of the boxes.

```
@BP @MK 4
```



Persistent:

No

Also See:

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

Notes:

The default marker shape for markers on box plots are: Series 1=Rectangle, Series 2=Circle, Series 3=Triangle "tip up", Series 4=Diamond, Series 5=Five Pointed Star, Series 6=Thin Plus Sign, Series 7=X-Shape, and Series 8=Triangle "tip down". These default marker shapes can be changed using the "@MS" macro.

@MS (Marker Shapes for Box Plots)

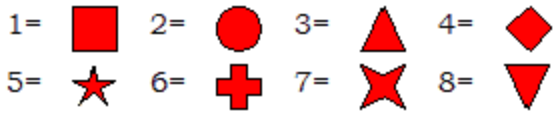
For Box Plots only, this macro sets the shape of markers for a particular series in a chart.

Syntax:

```
@MS n1 n2
```

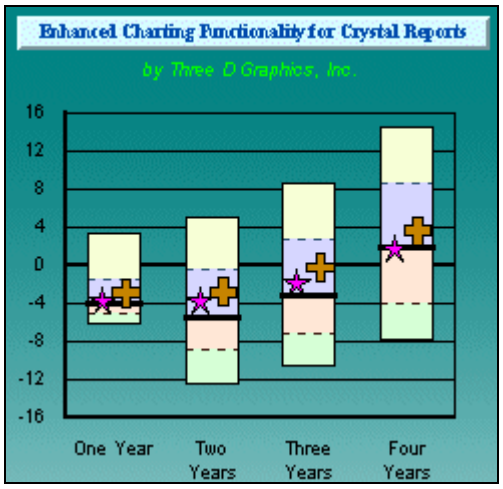
Parameters:

n1; Series Number (1...8).
n2; 1...8 selects the shape to assign to series *n1*:



Example:

```
@BP
@MK 2
@SZ 24
@MS 1 5
@MS 2 6
```



Persistent:

No

Notes:

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

@NEG_STYLE (Negative Value Style)

This macro sets the format of all numeric labels in a chart that have negative values.

Syntax:

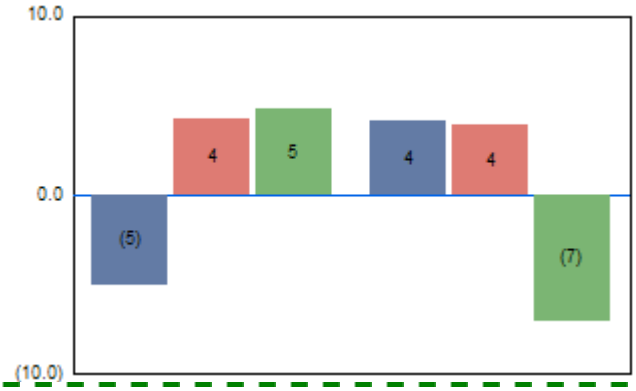
```
@NEG_STYLE nParenthesis
```

Parameters:

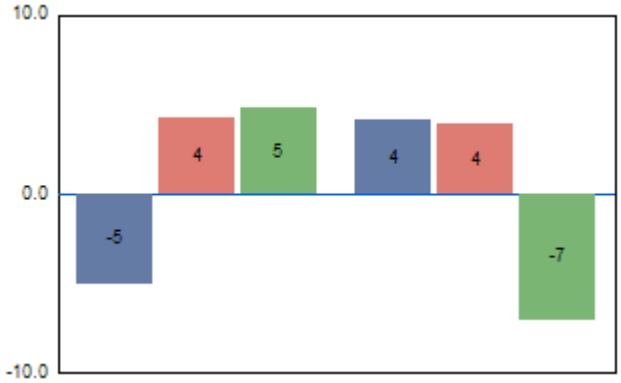
parentheses; 1 = format negative values with parenthesis (e.g., (1)), 0 = format negative values with a minus sign (e.g., -1)

Examples:

```
@NEG_STYLE 1
```




```
@NEG_STYLE 0
```



Persistent:

Yes

@PARAM_FIXUP (Parameter Fix-Up)

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

Syntax:

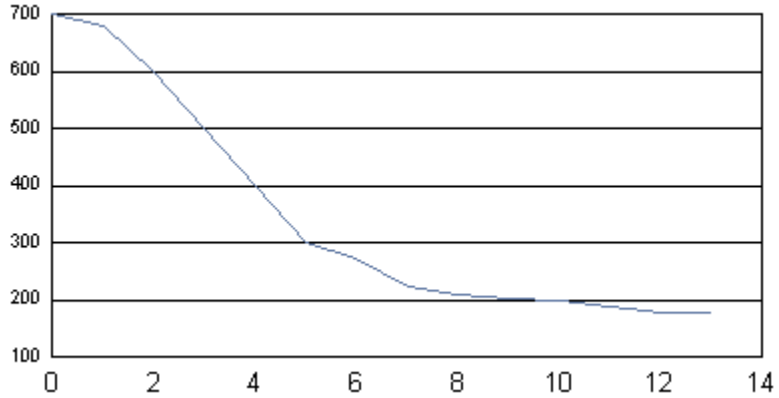
```
@PARAM_FIXUP nIndex
```

Parameters:

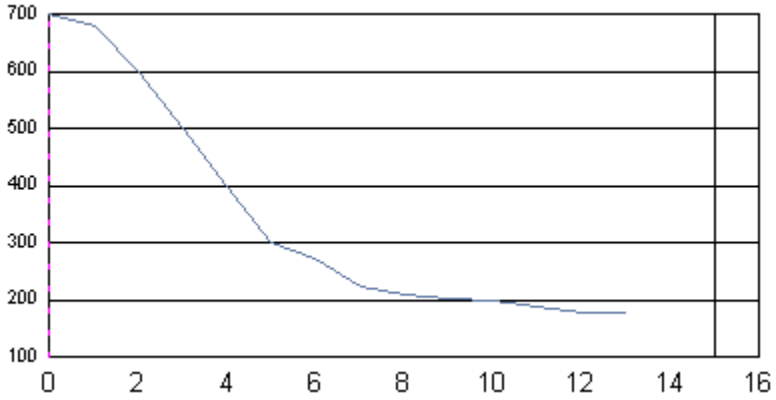
nIndex; 1=Enable parameter fix-up, 0=Disable parameter fix-up

Examples:

```
@X P1
@PARAM_FIXUP 0
```




```
@X P1
@PARAM_FIXUP 1
```



In this example, the parameter-driven user-defined line (@X P1) only appears when the @PARAM_FIXUP 1 macro is applied.

Persistent:

No

@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:

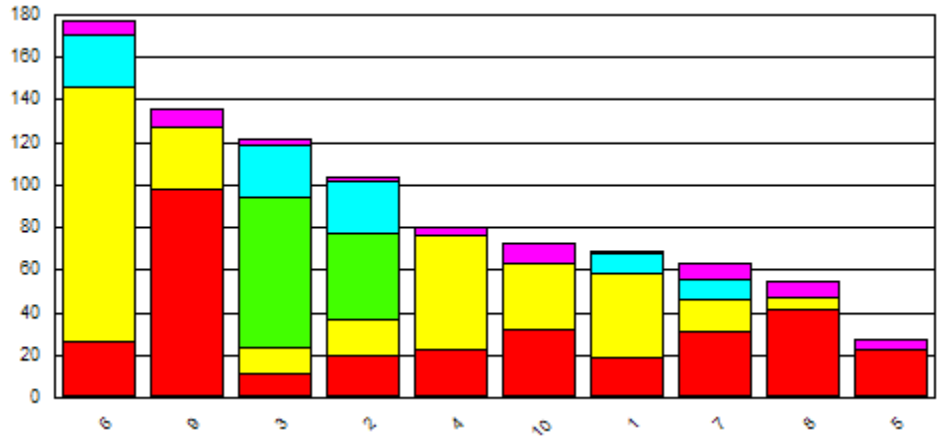
```
@PARETO n
```

Parameters:

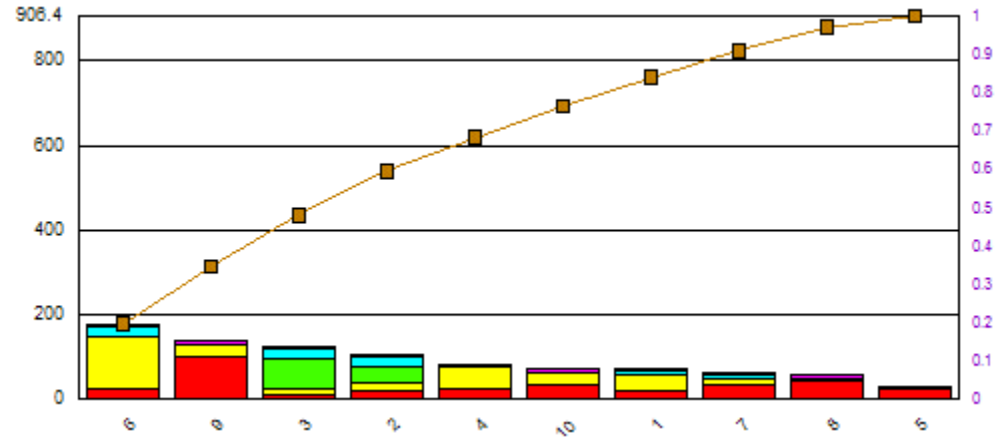
n; 0 = create a simple pareto chart. 1 = create a "classic" pareto chart with a cumulative percentage line.

Example:

```
@PARETO 0
```




```
@PARETO 1
```



Persistent:

Yes

@PAT (Riser Pattern)

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

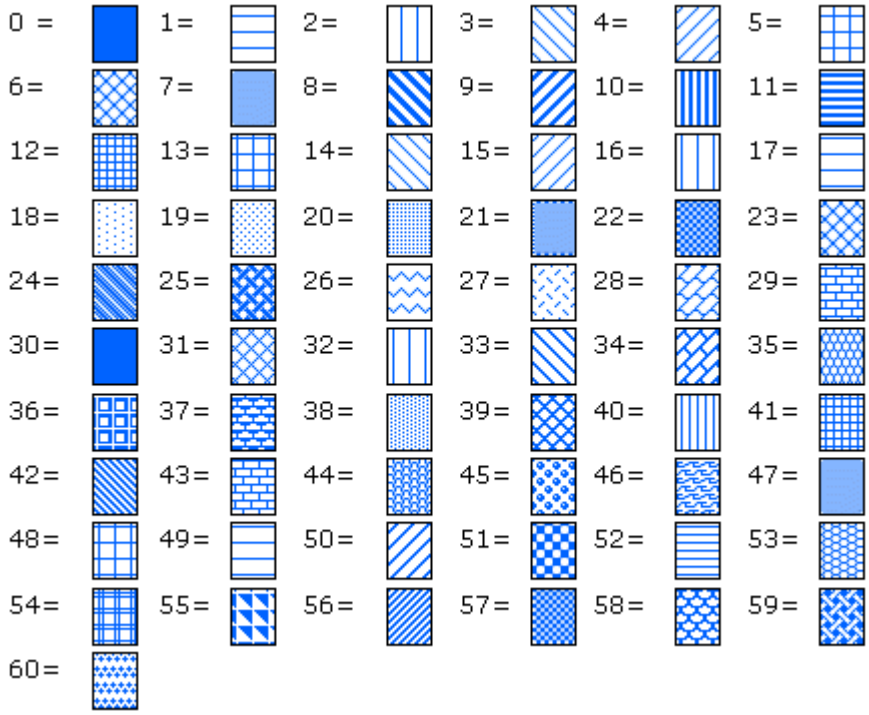
Syntax:

```
@PAT nSeries nPattern
```

Parameters:

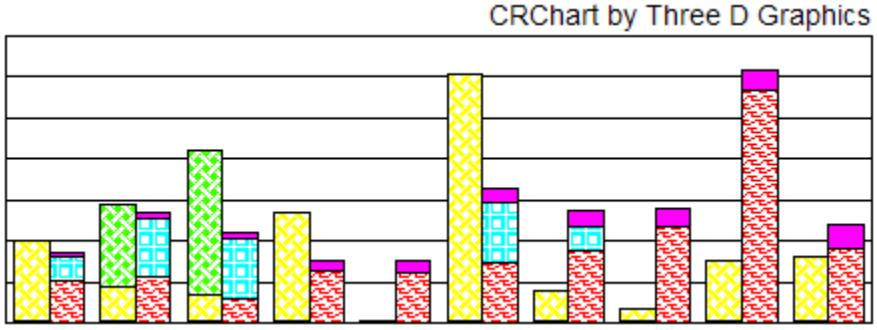
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

nPattern; 0...60 selects one of the following patterns with a white background. You can use –1 to –6 to select one of these patterns with a transparent background.



Example:

```
@PAT 1 46 @PAT 2 25 @PAT 3 59 @PAT 4 36
```



Persistent:

Yes

@PIE_NEG (Negative Values in Pie Legend)

This macro enables handling of negative values that may be shown in the legend area of a pie chart.

Syntax:

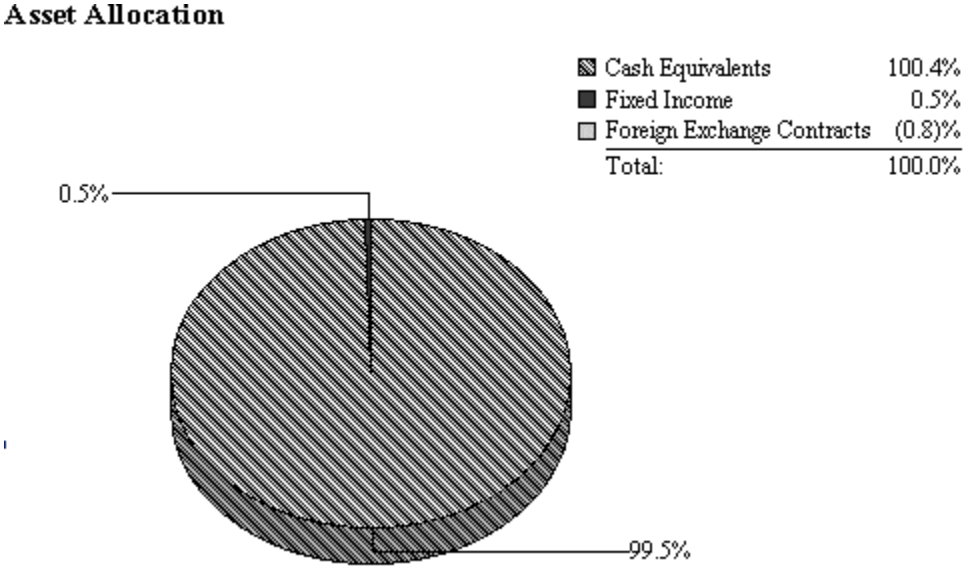
```
@PIE_NEG
```

Parameters:

None

Example:

```
@PIE_NEG
```



Notes:

The legend in a pie chart is enabled in the Look section of the Chart Options dialog. In addition to Show Legend, you can select Show Values and/or Show Percentage. These selections will show the total value and/or percentage that each pie section represents.

Persistent:

Yes

@POLAR (Polar Chart)

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

Syntax:

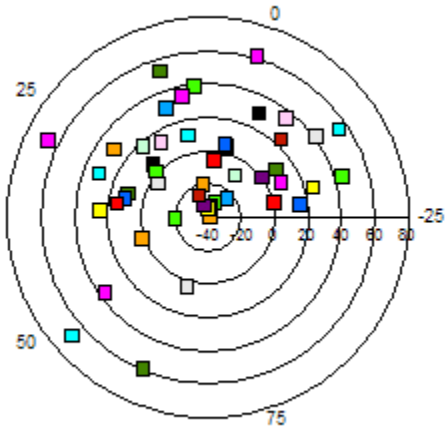
```
@POLAR
```

Parameters:

None

Example:

```
@POLAR
```



Persistent:

Yes

@PX (Precision on X-Axis)

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

Syntax:

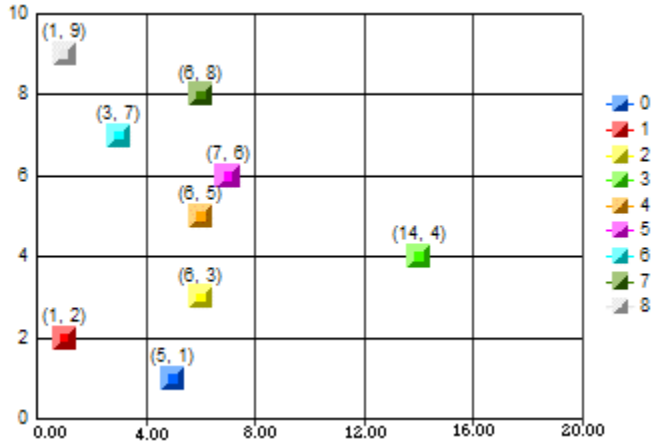
```
@PX n
```

Parameters:

n; Number of decimal places to use

Example:

```
@PX 2
```



Persistent:

Yes

Also See:

@PY

@PY (Precision on Y-Axis)

This macro sets the decimal precision to be used on the Y-axis.

Syntax:

```
@PY n
```

Parameters:

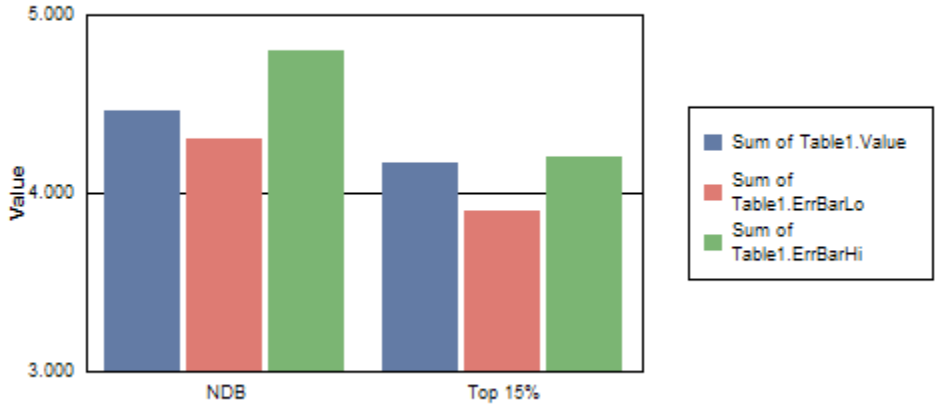
n; Number of decimal places to use

Example:

This example sets the number of decimal places to be shown on the Y-axis to 3.

```
@PY 3
```

Figure 1 - Average of All Patient Ratings



Persistent:

Yes

Also See:

@PX

@RESET (Reset Internal Data Range)

This macro is for debugging purposes only. 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

@RG (Reverse Group)

This macro reverses the order of groups in a chart.

Syntax:

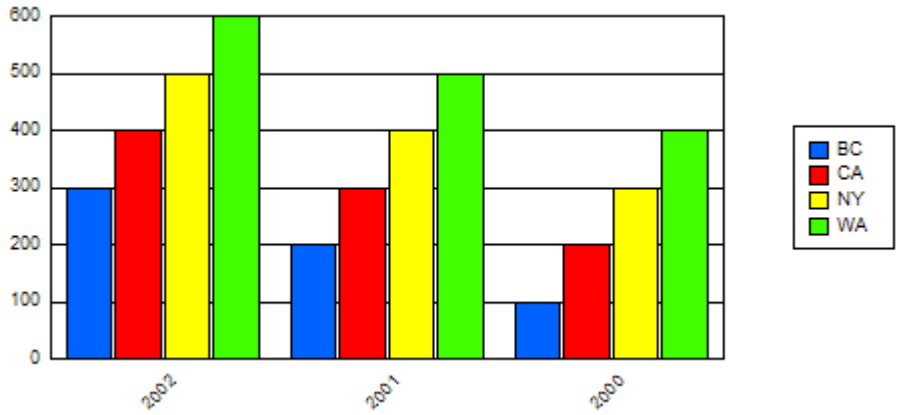
```
@RG n
```

Parameters:

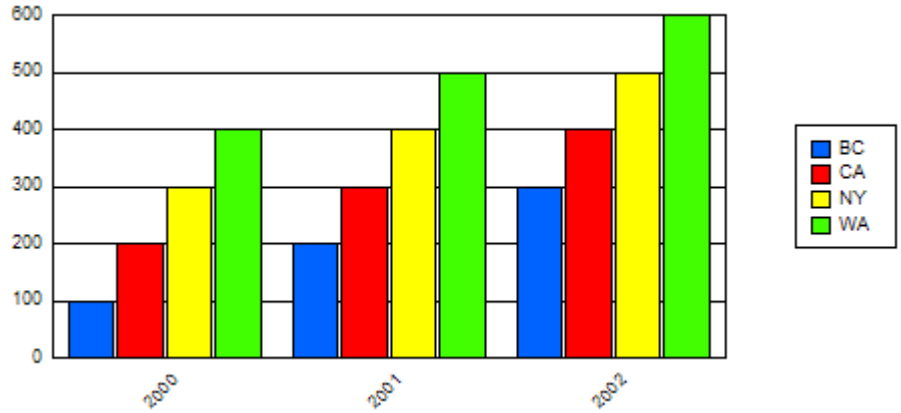
n; 1=true/reverse groups, 0=false/do not reverse groups.

Example:

```
@RG 1
```




```
@RG 0
```



Persistent:

Yes

Also See:

@RS

@RISER_BORDER (Riser Border)

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

Syntax:

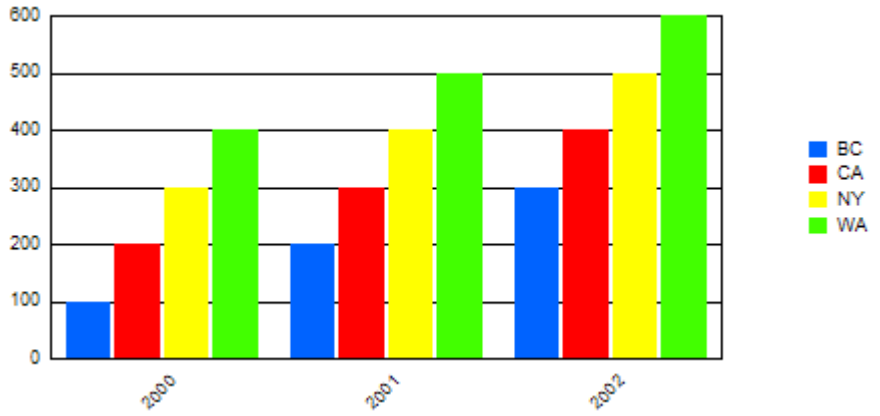
```
@RISER_BORDER bShow
```

Parameters:

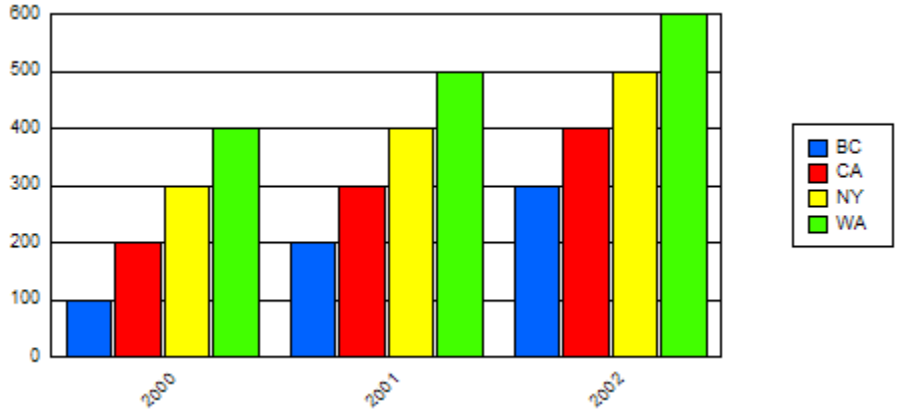
bShow; 0 = Turn OFF border line on risers/markers, 1= Turn ON border line on risers/markers.

Example:

```
@RISER_BORDER 0
```




```
@RISER_BORDER 1
```



Persistent:

Yes

@RISER_WIDTH (Riser Width)

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

Syntax:

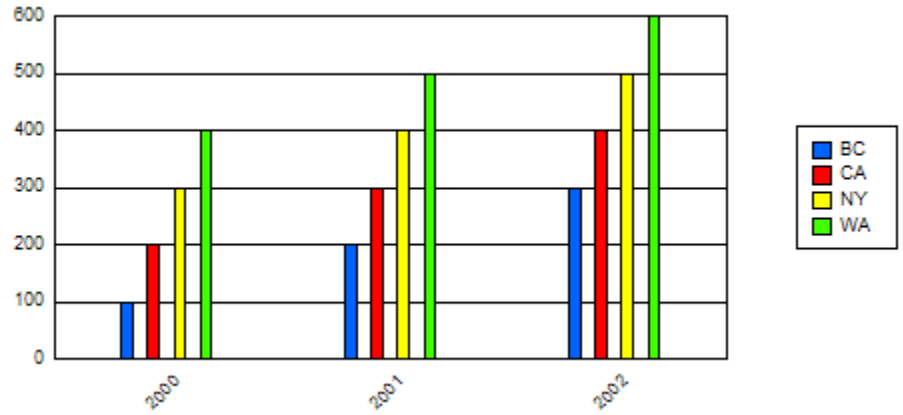
```
@RISER_WIDTH nWidth
```

Parameters:

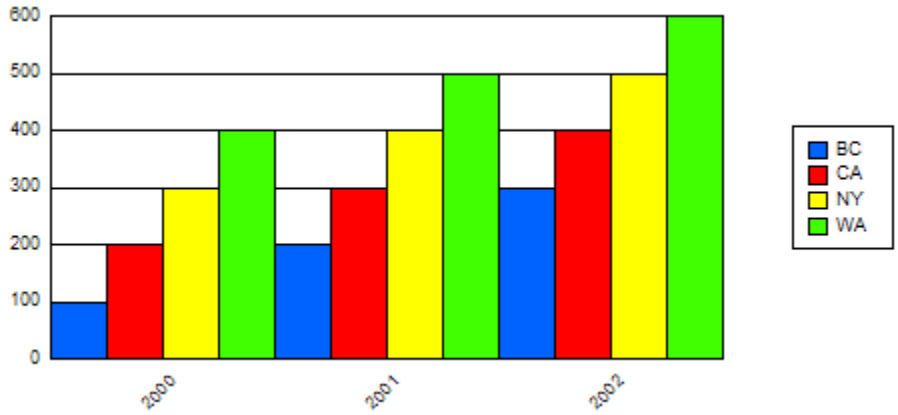
nWidth; 0...100 selects the width of risers.

Example:

```
@RISER_WIDTH 20
```




```
@RISER_WIDTH 100
```



Persistent:

Yes

@RS (Reverse Series)

This macro reverses the order of series in a chart.

Syntax:

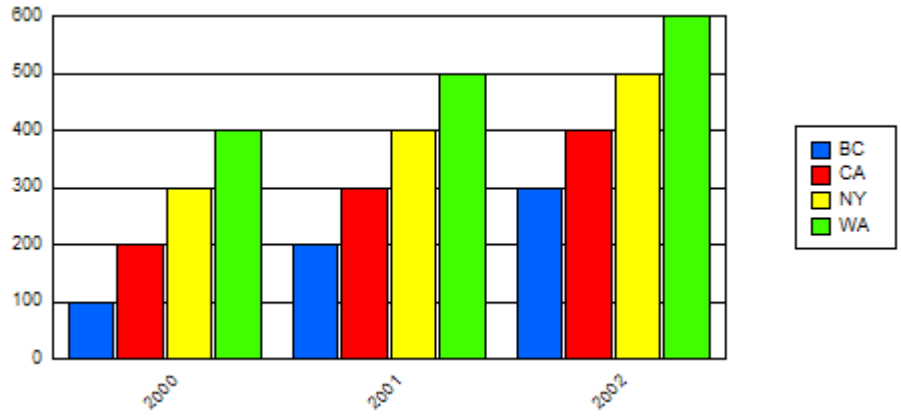
```
@RS n
```

Parameters:

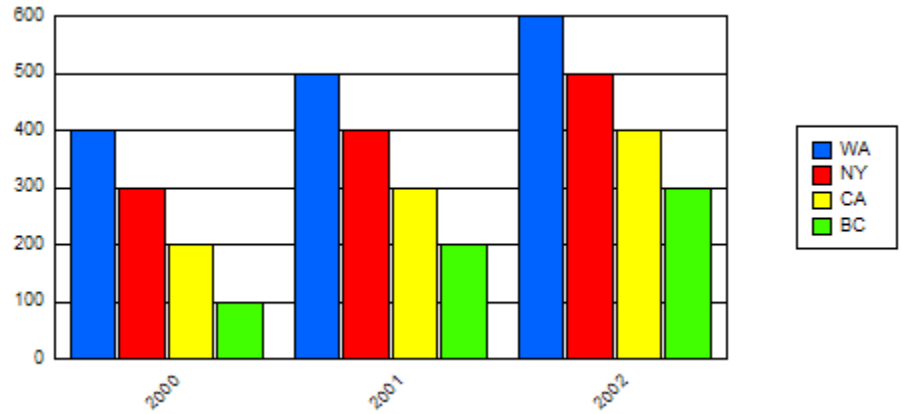
n; 1=true/reverse series, 0=false/do not reverse series.

Example:

```
@RS 0
```




```
@RS 1
```



Persistent:

Yes

Also See:

@RG

@SC (Scale on Y-Axis)

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

Syntax:

```
@SC f1 f2
```

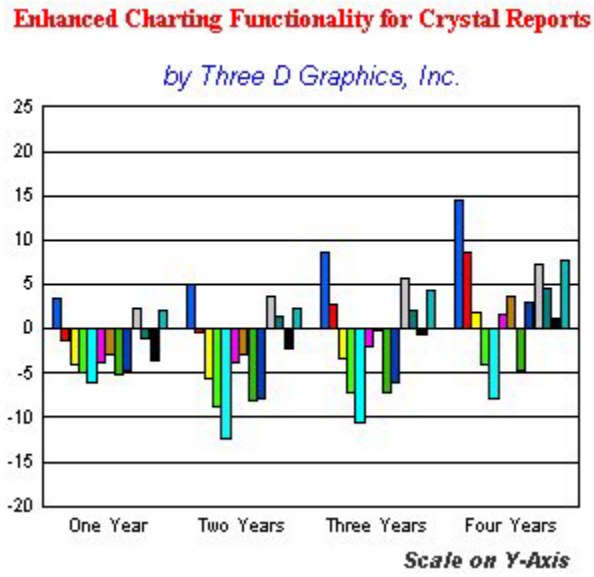
Parameters:

f1; Minimum value to show on the Y-Axis
f2; Maximum value to show on the Y-Axis

Example:

This example sets the Y-Axis scale to go from a minimum value of –20.0 to a maximum value of 25.0

```
@SC –20 25
```



Persistent:

Yes

Also See:

@SCX

@SCX (Scale on X-Axis)

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 f1 f2
```

Parameters:

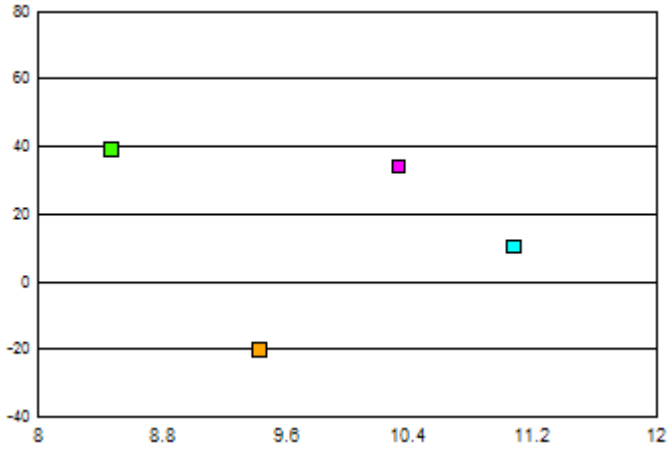
f1; Minimum value to show on the X-Axis

f2; Maximum value to show on the X-Axis

Example:

This example sets the minimum value (8) and maximum value (12) to be shown on the X-axis.

```
@SCX 8 12
```



Persistent:

Yes

Also See:

@SC

@SCY2 (Scale on Y2-Axis)

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

Syntax:

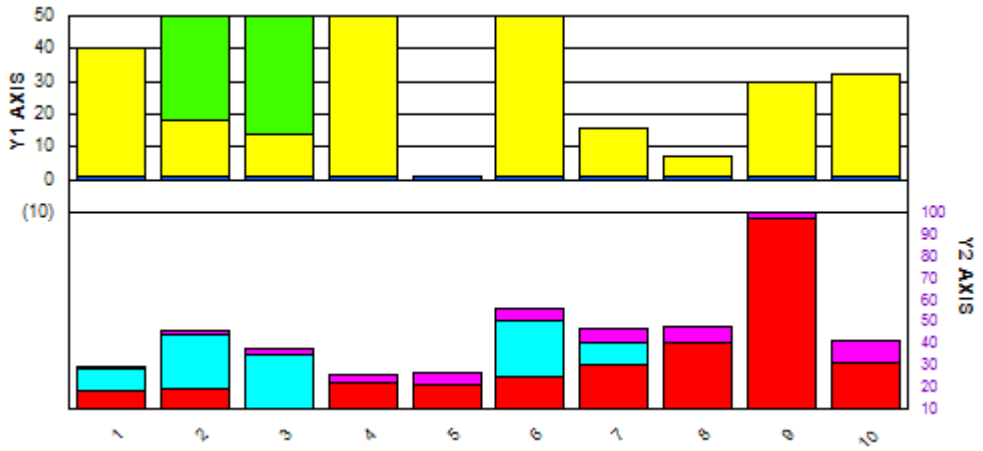
```
@SCY2 f1 f2
```

Parameters:

f1; Minimum value to show on the Y2-Axis
f2; Maximum value to show on the Y2-Axis

Example:

```
@SC -10 50
@SCY2 10 100
@GRAPHTYPE
```



Persistent:

Yes

@SHADOW (Drop Shadow)

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

Syntax:

```
@SHADOW nObject nXOffset nYOffset
```

Parameters:

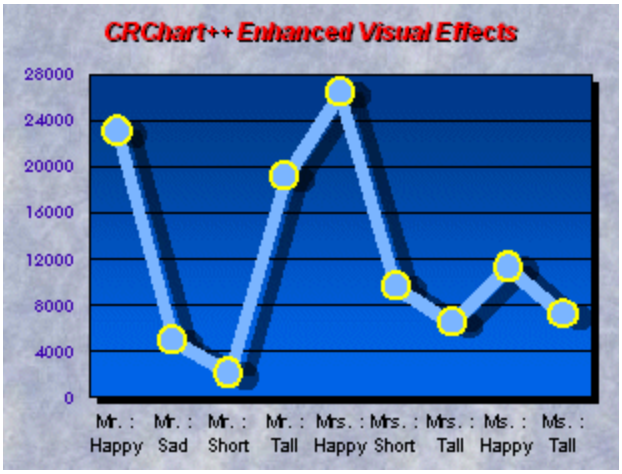
nObject; 0...12 selects one of the following objects:

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 @SHADOW 0 200 400
```



Persistent:

Yes

@SMOOTH_LINE (Smooth/Straight Lines)

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

Syntax:

```
@SMOOTH_LINE nSeries bOnOff
```

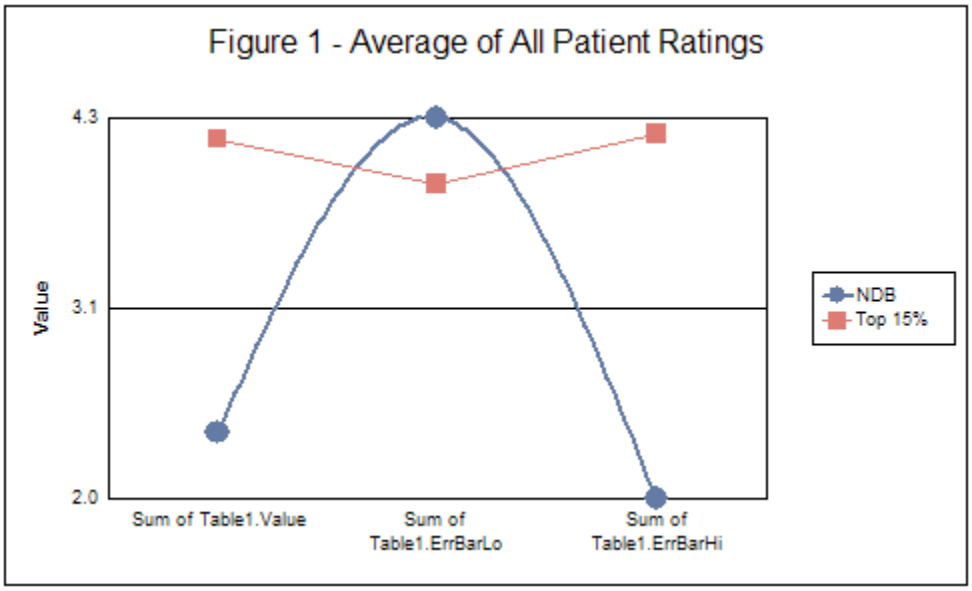
Parameters:

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:

```
@SMOOTH_LINE 0 1
```



Persistent:

Yes

@SORT (Sort Series/Groups)

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

Syntax:

```
@SORT n
```

Parameters:

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

1 = Sort Series in Reverse Alphabetical order

2 = Sort Series in Alphabetical order

3 = Sort Groups in Alphabetical order

4 = Sort Groups in Reverse Alphabetical order

5 = Sort Series in ascending numeric value (i.e. total all values in each series. Then 'rank' the series from smallest total to largest).

5 = Sort Series in descending numeric value

6 = Sort Groups in ascending numeric value

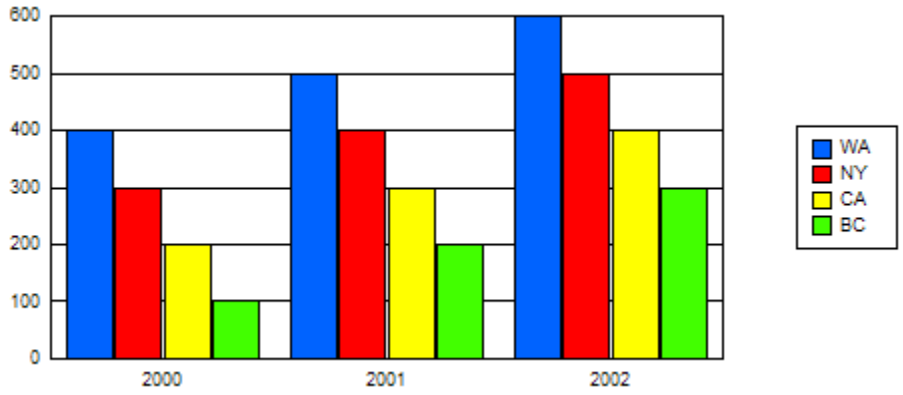
7 = Sort Groups in descending numeric value

8 = First Series Key to Groups Ascending

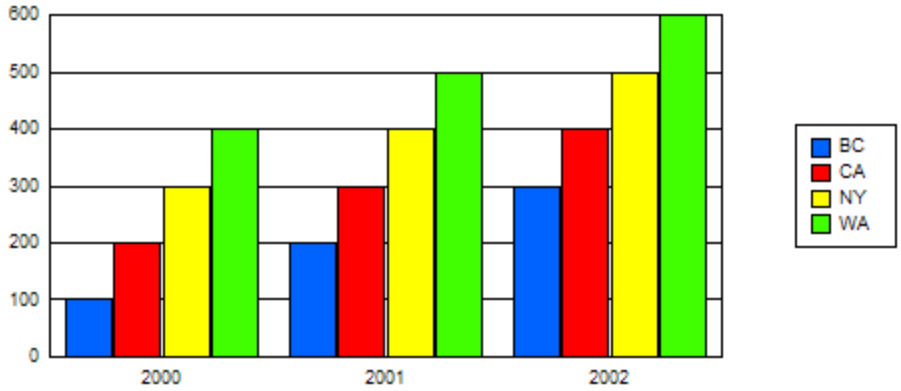
9 = First Series Key to Groups Descending

Examples:

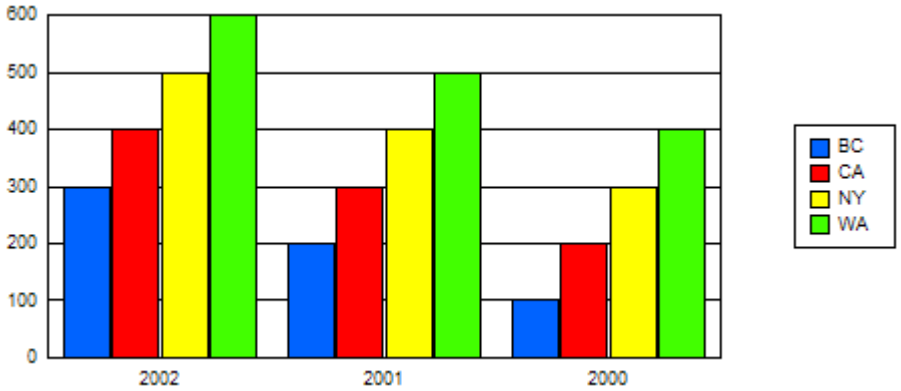
```
@SORT 1
```



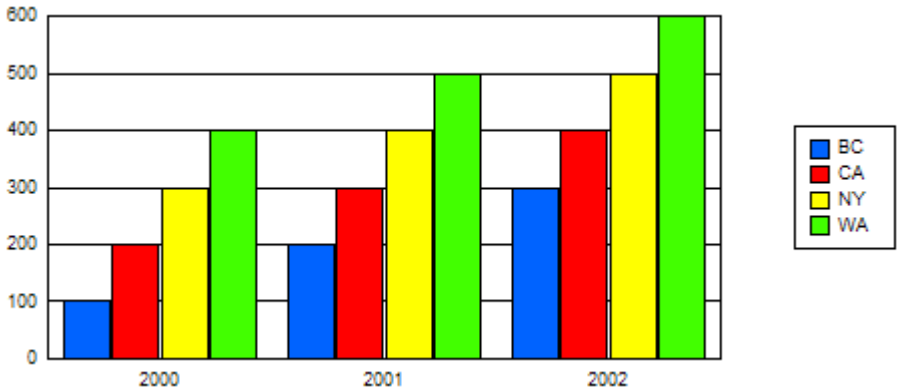

```
@SORT 2
```



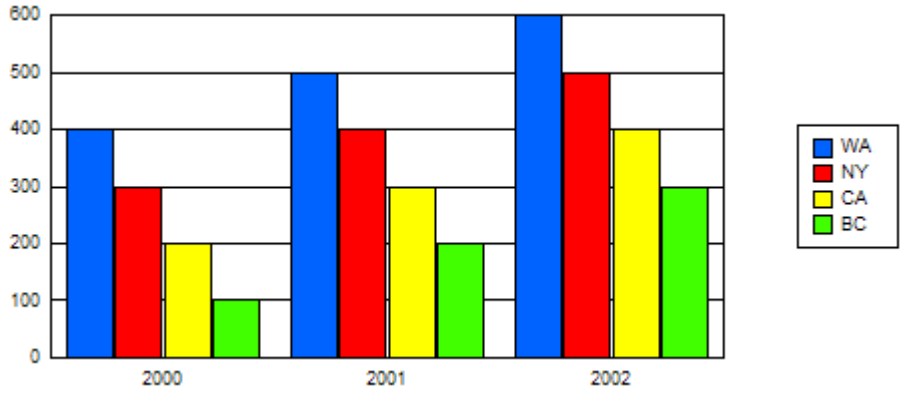

```
@SORT 3
```



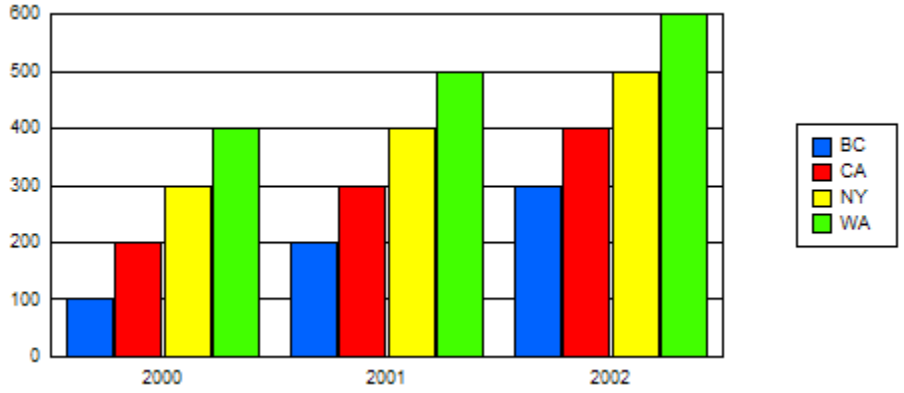

```
@SORT 4
```



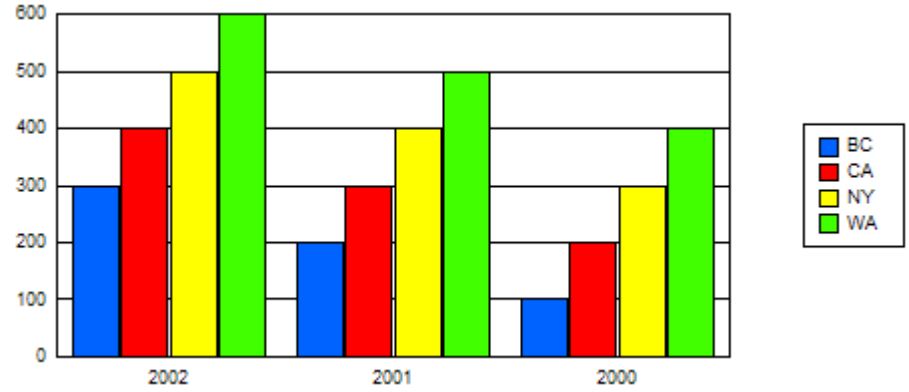

```
@SORT 5
```




```
@SORT 6
```




```
@SORT 7
```



Persistent:

No

@STOP

This macro sets the number of elements in a scatter chart that are force-assigned to the first series.

Syntax:

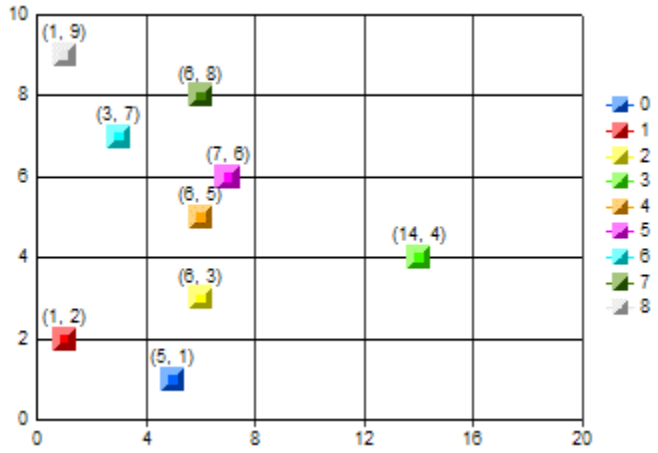
```
@STOP n
```

Parameters:

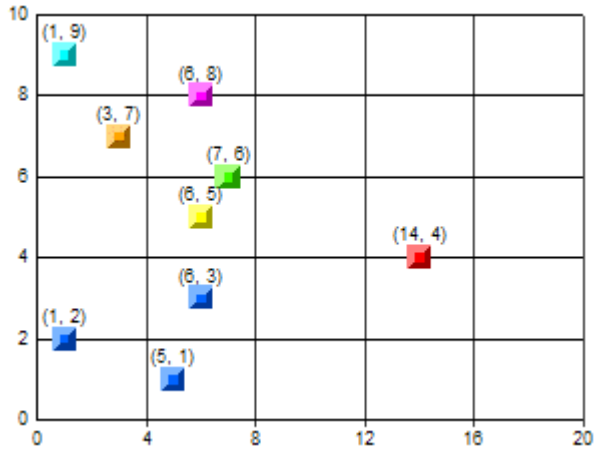
n; number of elements to force assign to first series.

Example:

Before @STOP applied




```
@STOP 3
```



Persistent:

No

@STRIP_ZERO (Strip Zero Values)

This macro will remove all risers/markers with a value of 0.0 after the specified series.

Syntax:

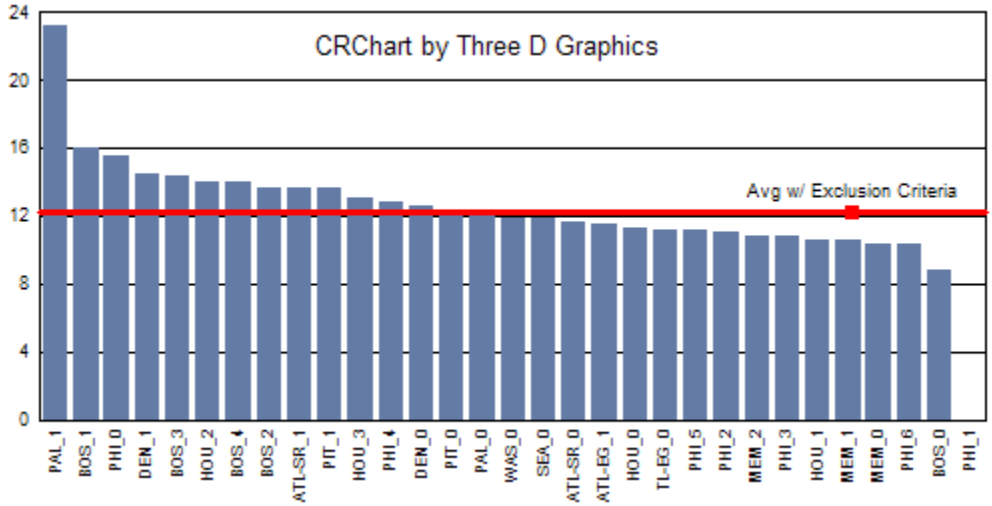
```
@STRIP_ZERO nSeries
```

Parameters:

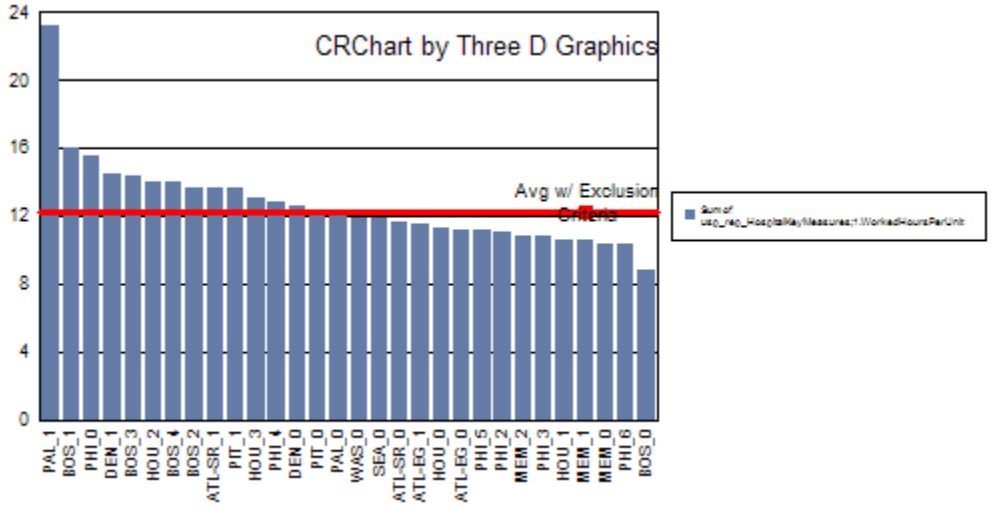
nSeries; Series Number (0…1024)

Examples:

```
Chart Before @STRIP_ZERO
```




```
@STRIP_ZERO 0
```



Persistent:

Yes

@SWAP (Swap Series/Groups)

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

Syntax:

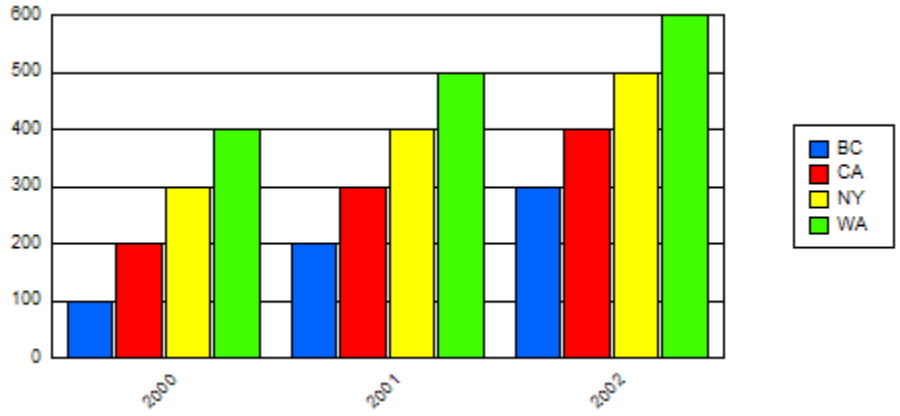
```
@SWAP n
```

Parameters:

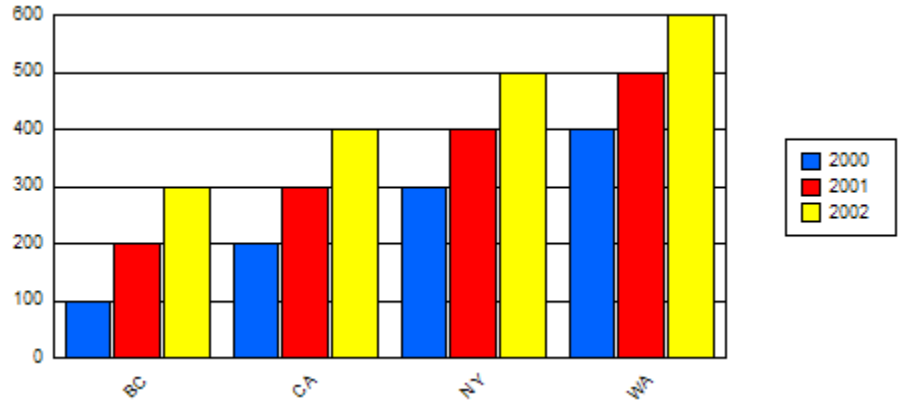
n; 1 = swap series/group orientation, 0= do not swap series/group orientation

Example:

```
@SWAP 0
```




```
@SWAP 1
```



Persistent:

Yes

@SZ (Size of Markers)

This macro sets the size of the markers in a BoxPlot or scatter chart.

Syntax:

```
@SZ n1
```

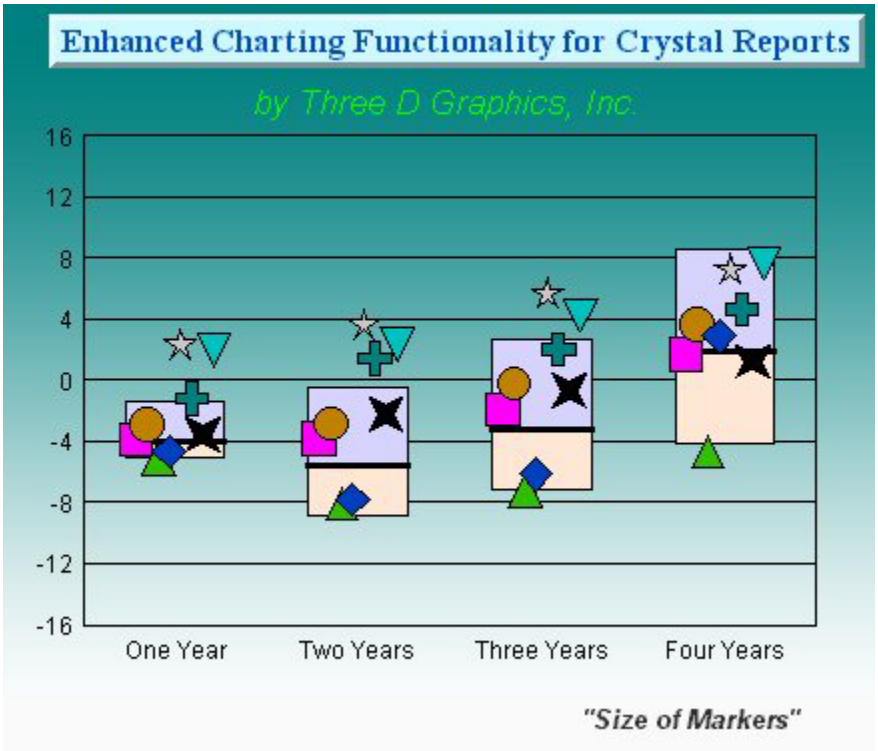
Parameters:

n1; Size of markers (1...100)

Example:

This example creates a box plot chart (@BP) with eight markers on each box (@MK 8). @SZ 50 sets the size of each marker.

```
@BP @MK 8 @SZ 50
```



Persistent:

No

Also See:

@MS & @MC

@TIMEAXIS (Time Axis)

This macro creates a time axis in bar, line, and area charts.

Syntax:

```
@TIMEAXIS fStart fStop fInterval
```

Parameters:

fStart; Start point in time. It must be specified in 1901 date format as defined in Microsoft EXCEL.

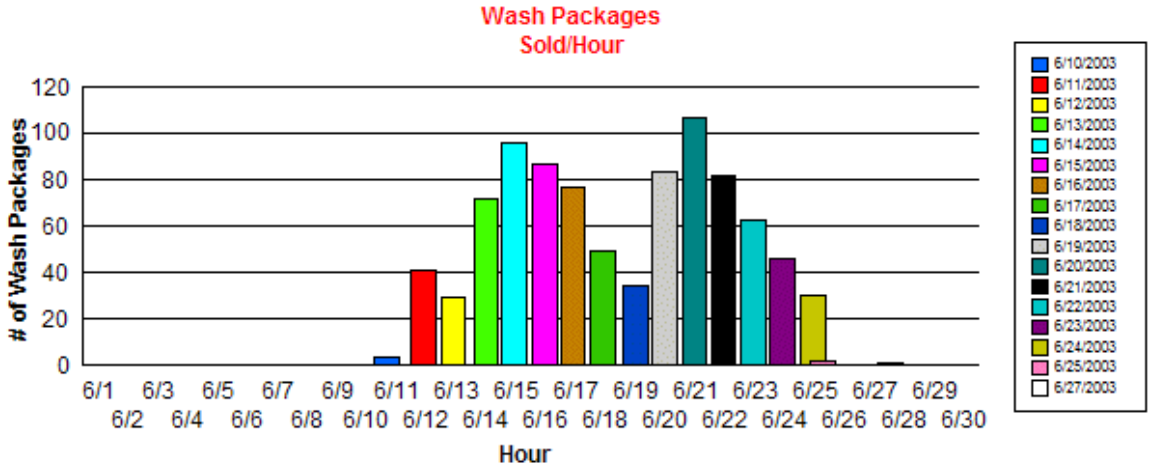
fStop; End point in time. It must be specified in 1901 date format as defined in Microsoft EXCEL.

fInternval; Amount of time between labels on axis. 1.0 = 1 day.

Example:

This example sets the start date (*fStart*) to June 1, the stop date (*fStop*) to June 30, and the interval (*fInterval*) to 1 day.

```
@TIMEAXIS 37773 37802 1
```



Persistent:

Yes

Notes:

This macro only works with the CR9.x and CR10.x replacement libraries. It will not work in the CR8.x replacement library!

@USER_CIRCLE (User Circle)

This macro draws a circle in a portion of the chart frame on 2D charts with an optional phrase. 0,0 is the lower left corner of the chart frame.

Syntax:

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

Parameters:

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

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

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

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

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255; defines the color of the line that is used to draw the circle.

nLineStyle; 0...15 Line Style (See @LS for a description of these values)

nLineThickness; 0...1000 Line Thickness

szPhrase; Optional phrase. It must be terminated with a '~'

Example:

```
See @USER_FILL_CIRCLE
```

Persistent:

No

Also See:

@USER_FILL_CIRCLE, @USER_FILL_CIRCLE2

@USER_FILL (User Rectangle, Color Fill)

This macro fills a portion of the chart frame on 2D charts with a particular color and a phrase. 0,0 is the lower left corner of the chart frame.

Syntax:

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

Parameters:

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

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

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

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

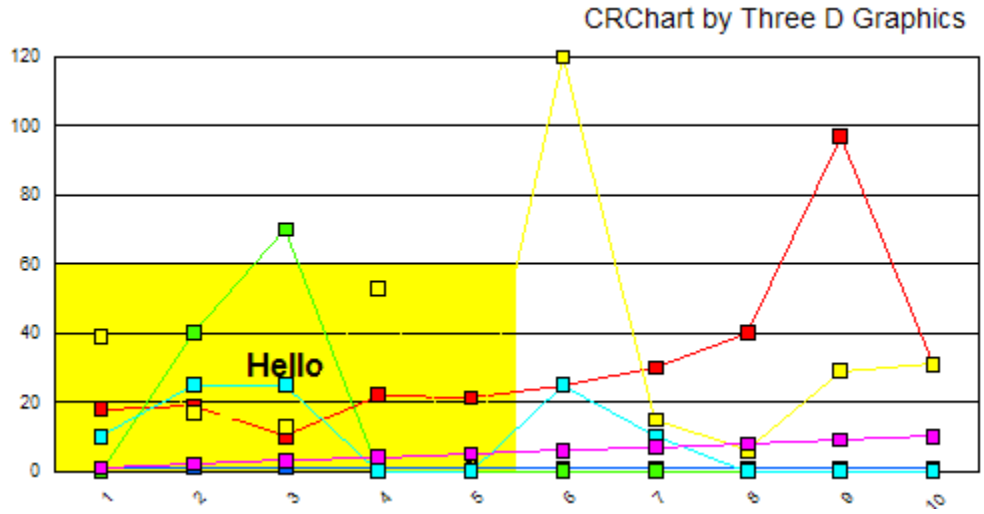
nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

szPhrase; Optional phrase. It must be terminated with a '~'

Example:

This example fills the lower left 'quadrant' of the chart frame [0.0 0.5 .0. 0.5] with yellow [255 255 0], and draws the phrase "Hello" [Hello~] in the middle.

```
@USER_FILL 0.0 0.5 0.0 0.5 255 255 0 Hello~
```



Persistent:

No

Also See:

@USER_FILL2, @USER_RECT

Notes:

A maximum of 5 @USER_FILL, @USER_FILL2, and/or @USER_RECT macros may be specified but any combination of the three macros may be used. @USER_FILL, @USER_FILL2, and @USER_RECT are independent of the X or Y axis values/scales.

@USER_FILL_CIRCLE (Use Circle, Color Fill)

This macro draws a circle in a portion of the chart frame on 2D charts with a particular color and an optional phrase. 0,0 is the lower left corner of the chart frame.

Syntax:

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

Parameters:

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

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

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

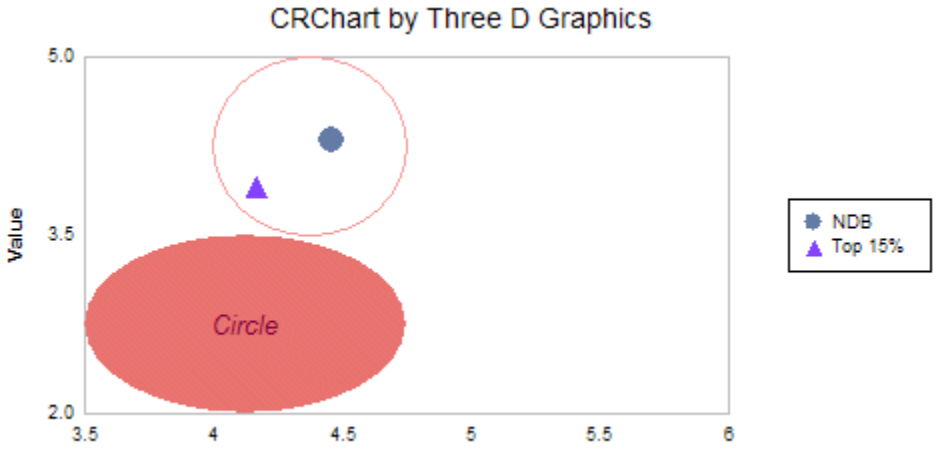
fStopY; 0.0...1.1 Y-Axis stop location

nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255; defines the color to fill the circle.

szPhrase; Optional phrase. It must be terminated with a '~'

Example:

```
@USER_FILL_CIRCLE 0 .5 0 .5 233 112 112 Circle~
@USER_CIRCLE 0.2 .5 .5 1 243 155 155 0 0
```



Persistent:

No

Also See:

@USER_CIRCLE, @USER_FILL_CIRCLE2

@USER_FILL_CIRCLE2 (User Circle, Pattern Fill)

This macro draws a circle in a portion of the chart frame on 2D charts with a particular color pattern and a phrase. 0,0 is the lower left corner of the chart frame.

Syntax:

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

Parameters:

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

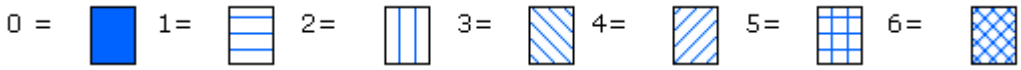
fStopX; 0.0...1.1 X-Axis stop location

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

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

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

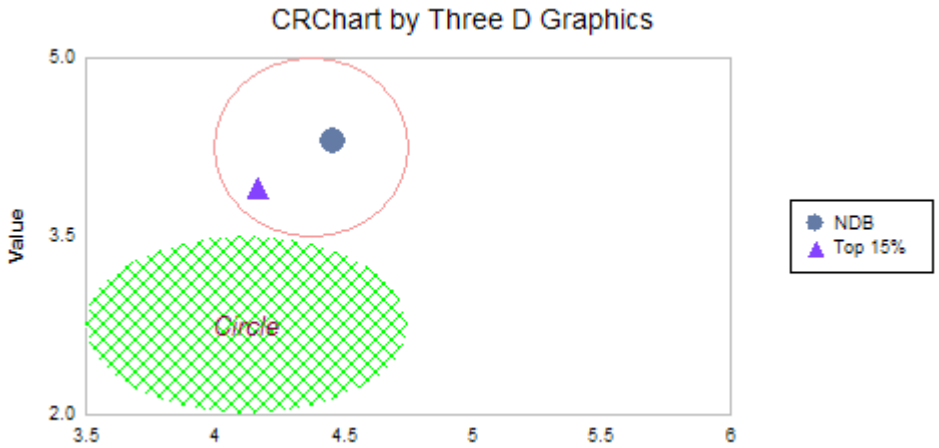
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



szPhrase; Optional phrase. It must be terminated with a '~'

Example:

```
@USER_FILL_CIRCLE2 0 .5 0 .5 0 255 0 6 Circle~
@USER_CIRCLE 0.2 .5 .5 1 243 155 155 0 0
```



Persistent:

No

Also See:

@USER_CIRCLE, @USER_FILL_CIRCLE

@USER_FILL2 (User Rectangle, Pattern Fill)

This macro fills a portion of the chart frame on 2D charts with a particular color pattern and a phrase. 0,0 is the lower left corner of the chart frame.

Syntax:

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

Parameters:

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

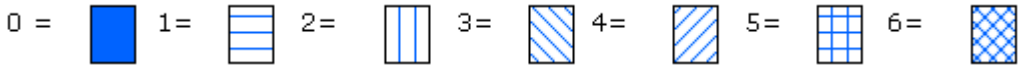
fStopX; 0.0...1.1 X-Axis stop location

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

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

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

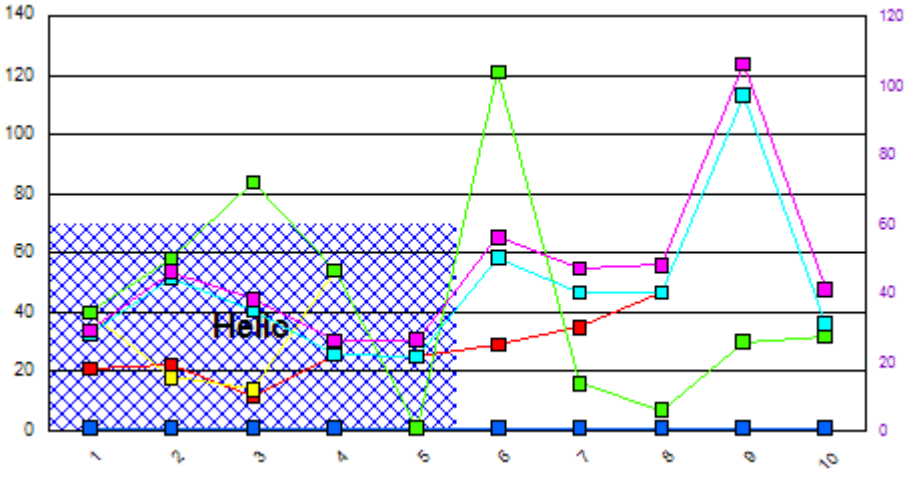
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



szPhrase; Optional phrase. It must be terminated with a '~'

Example:

```
@USER_FILL2 0.0 0.5 0.0 0.5 0 0 255 6 Hello~
```



Persistent:

No

Notes:

A maximum of 5 @USER_FILL, @USER_FILL2, and/or @USER_RECT macros may be specified but any combination of the three macros may be used. @USER_FILL, @USER_FILL2, 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 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.

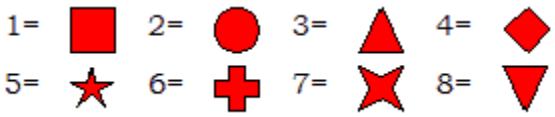
Syntax:

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

Parameters:

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

nShape; 1...8 selects one of the following markers:



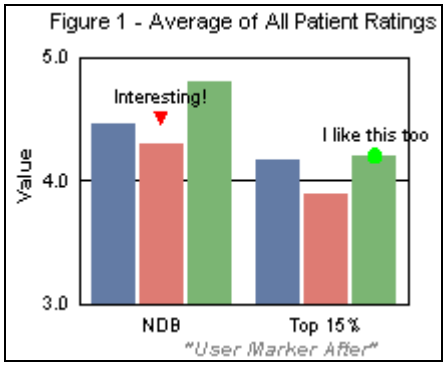
nRed; 0..255, *nGreen;* 0..255, *nBlue;* 0..255 define the color of the marker.

phrase; Optional phrase. It must be terminated with a '~'

Example:

This example draws a red [255 0 0] Triangle "tip-down" [16] marker at the [1.0 4.5] xy coordinates with text [Interesting!~] and a bright green [0 255 0] circle [4] marker at the [2.3 4.2] xy coordinates with text [I like this too~].

```
@USER_MARKER 1.0 4.5 16 255 0 0 Interesting!~
@USER_MARKER 2.3 4.2 4 0 255 0 I like this too~
```



Persistent:

No

Notes:

Only 8 USER_MARKERs may be specified.

@USER_RECT (User Rectangle)

This macro fills a portion of the chart frame on 2D charts with a particular color and a phrase. 0,0 is the lower left corner of the chart frame.

Syntax:

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

Parameters:

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

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

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

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

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

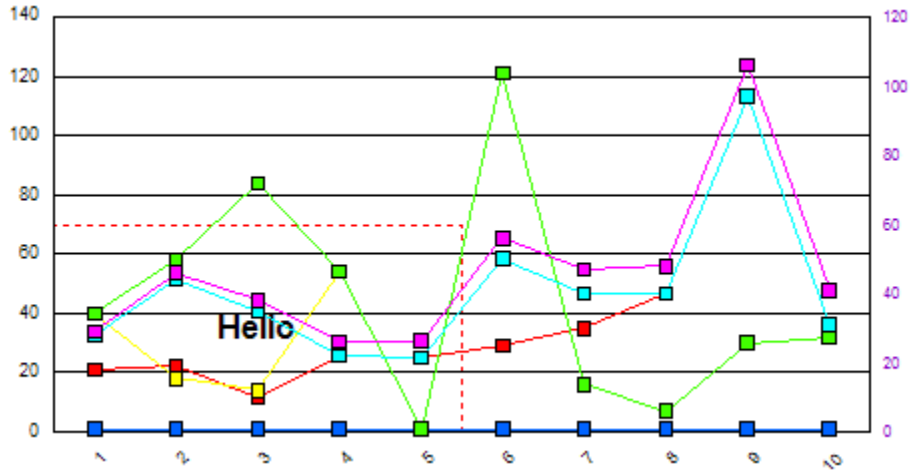
nLineStyle; 0...15 Line Style (See @LS for a description of these values)

nLineThickness; 0...1000 Line Thickness

szPhrase; Optional phrase. It must be terminated with a '~'

Example:

```
@USER_FILL 0.0 0.5 0.0 0.5 255 0 0 2 8 Hello~
```



Persistent:

No

Also See:

@USER_FILL, @USER_FILL2

Notes:

A maximum of 5 @USER_FILL, @USER_FILL2, and/or @USER_RECT macros may be specified but any combination of the three macros may be used. @USER_FILL, @USER_FILL2, and @USER_RECT are independent of the X or Y axis values/scales.

@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.

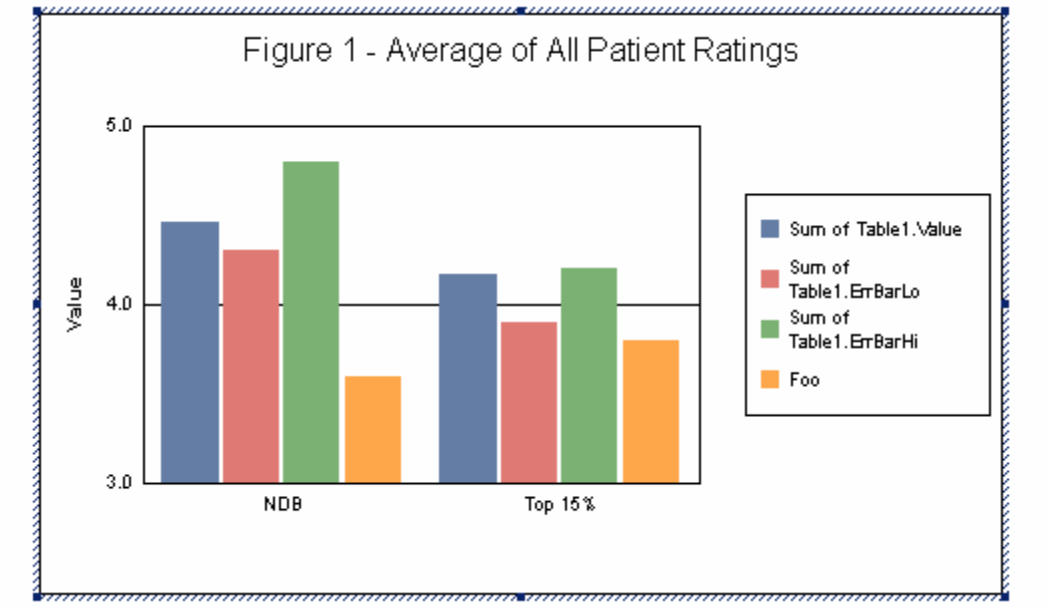
fValue1 fValue2 ... fValueN; Values to assigned to each *nElements.*

szSeriesName; Name of the new series what will appear in the legend.

Example:

This example adds a series called "Foo" with 2 values (3.6 and 3.8).

```
@USER_SERIES 2 3.6 3.8  Foo~
```



Persistent:

No

@WATERFALL (Waterfall Chart)

This macro creates a waterfall chart.

Syntax:

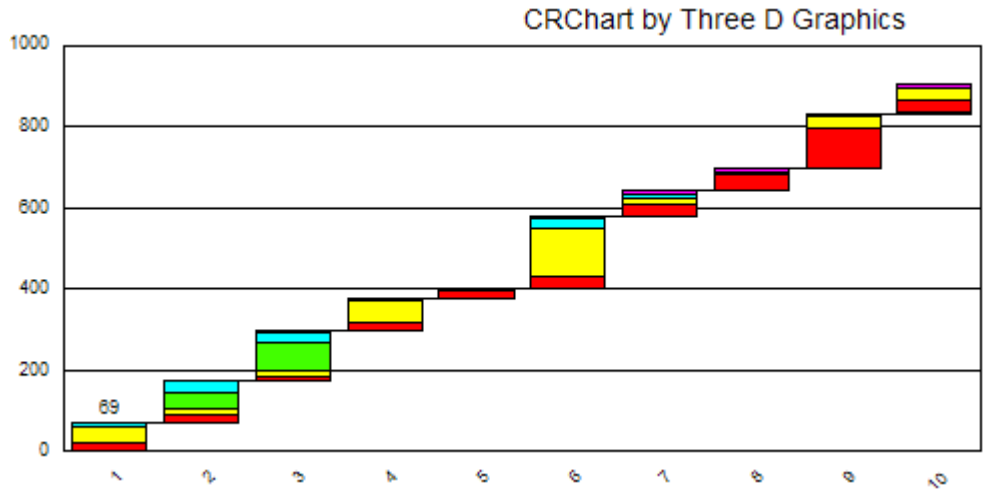
```
@WATERFALL
```

Parameters:

None

Example:

```
@WATERFALL
```



Persistent:

Yes

@X (User-Defined Vertical Line)

This macro adds a user-defined vertical 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 vertical line that is 50% of the way between the left and right sides of the chart frame.

Syntax:

```
@X n
```

Parameters:

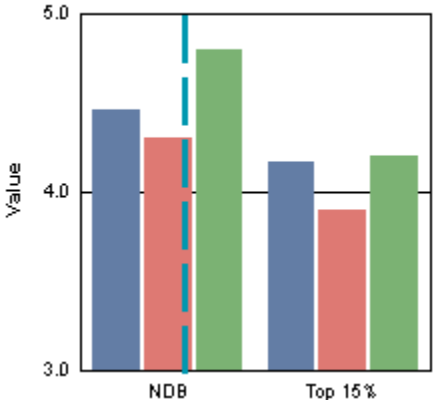
n; Value at which to add the user-defined line on the x-axis

Example:

This example adds a free line to the current chart. The line will be vertical and start on the x-axis at value .3.

```
@X .3
```

Figure 1 - Average of All Patient Ratings



Persistent:

No

@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:

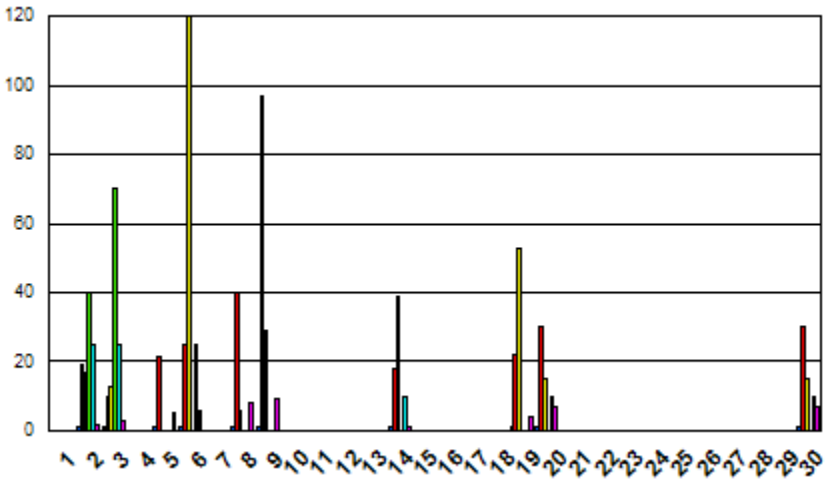
nMode; 0

nStart; Starting group label string

nStop; Ending group label string.

Example:

```
@X_AXIS_MODE 0 1 30
```



Notes:

1) In this version of CRChart, label strings must be digits (e.g., 1999, 2000, 2001, etc.) that can be converted to integer values. Future releases will support time labels (e.g., 8:00am), month-day labels (e.g., June 1, June 2) and day-of-week labels (e.g., Monday, Tuesday).

2) This macro will not work if group labels are aliased with the @AGL macro.

Persistent:

No

@XSKIP (Skip Labels on X-Axis)

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

Syntax:

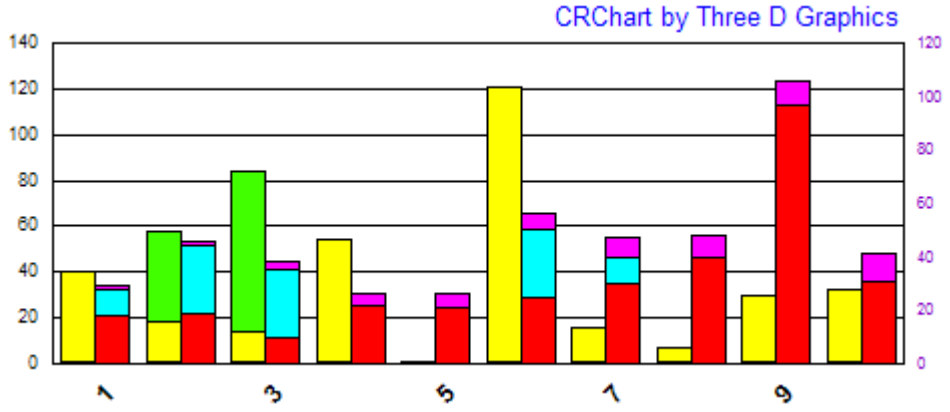
```
@XSKIP value
```

Parameters:

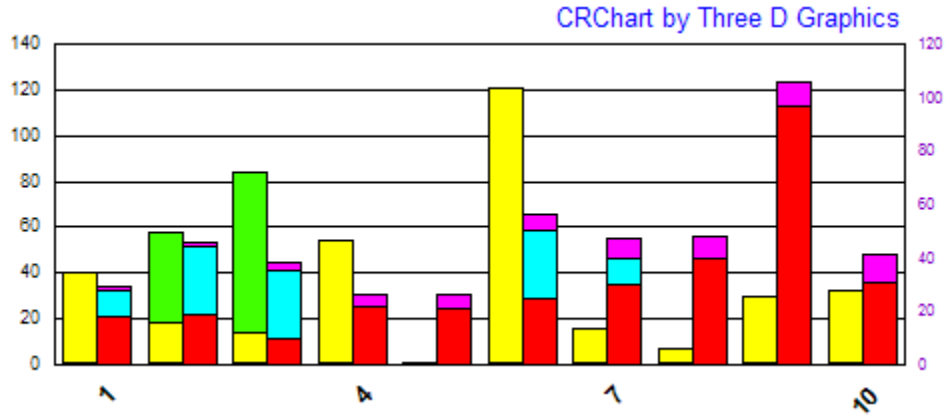
value; skip value

Example:

```
@XSKIP 2
```




```
@XSKIP 3
```



Persistent:

No

@XSKIP2 (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:

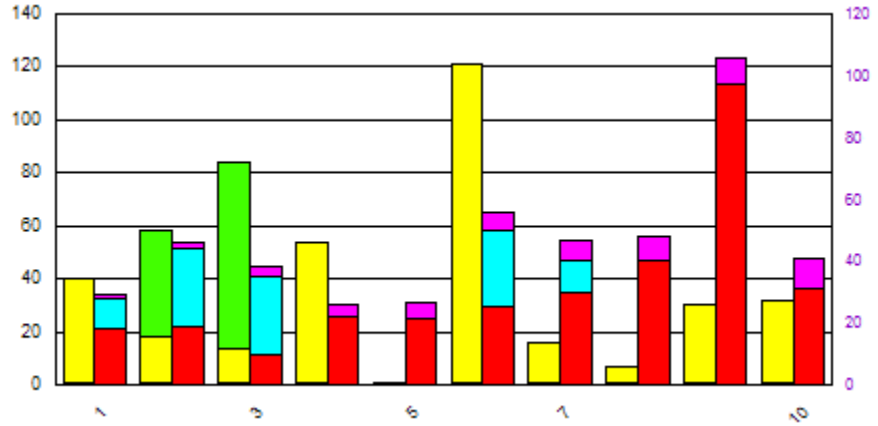
```
@XSKIP2 value
```

Parameters:

value; skip value

Example:

```
@XSKIP2 2
```



Persistent:

No

@XY (User-Defined 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

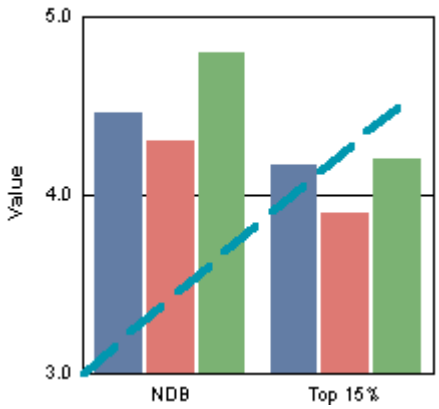
ny2; Ending y-coordinate

Example:

This example adds a free line to the chart. The line starts at x=0,y=3 and end at x=0.9, y=4.4.

```
@XY 0 3 0.9 4.5
```

Figure 1 - Average of All Patient Ratings



Persistent:

No

@XY_DP2 (Draw User 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:

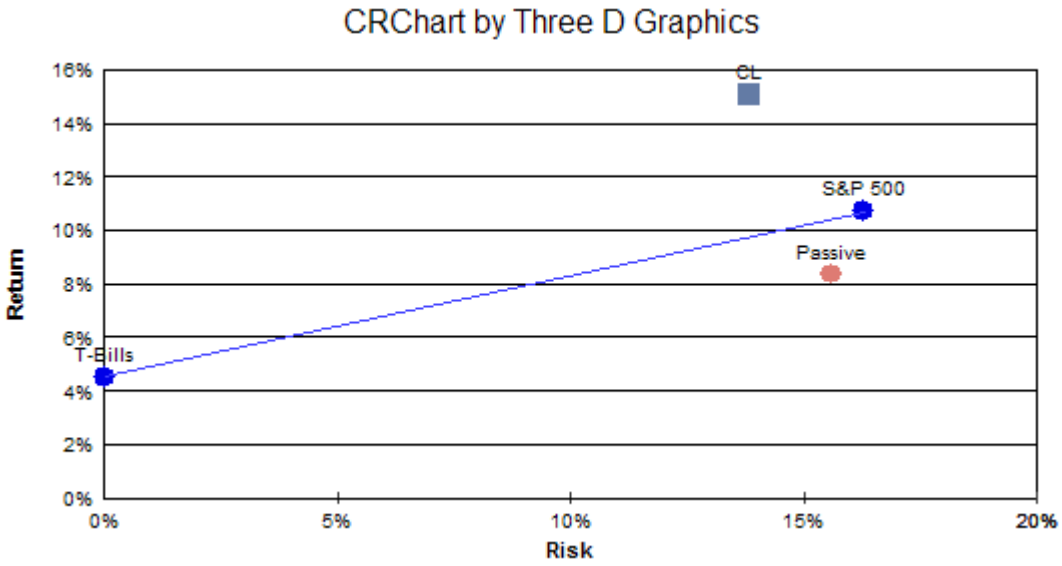
s1, g1; Zero-based series number and group number to start drawing line.

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

Example:

This example draws a line between the first point in series 2 and the first point in series 3.

```
@XY_DP2 2 0 3 0
```



Persistent:

No

@Y (User-Defined Horizontal Line)

This macro adds a user-defined line on the y-axis at value *n*.

Syntax:

```
@Y n
```

Parameters:

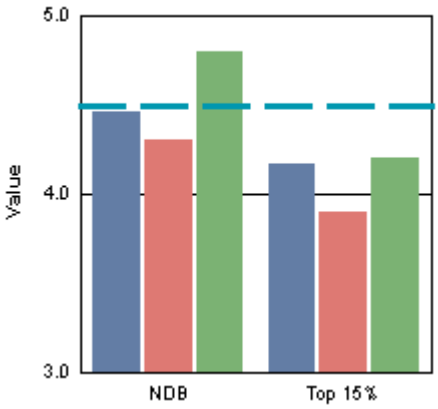
n; Value at which to add the user-defined line on the y-axis

Example:

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

```
@Y 4.5
```

Figure 1 - Average of All Patient Ratings



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 *f1* draw "Up" from the baseline. Values less than *f1* draw "Down" from the baseline.

Syntax:

```
@Y1BASE f1
```

Parameters:

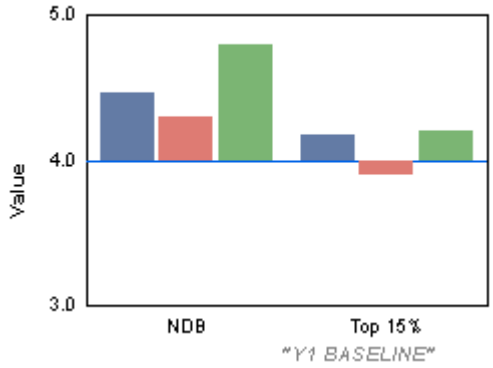
f1; Base line value

Example:

This example sets the base line for the Y1-axis to 4.

```
@Y1BASE 4
```

Figure 1 - Average of All Patient Ratings



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 *f1* draw "Up" from the baseline. Values less than *f1* draw "Down" from the baseline.

Syntax:

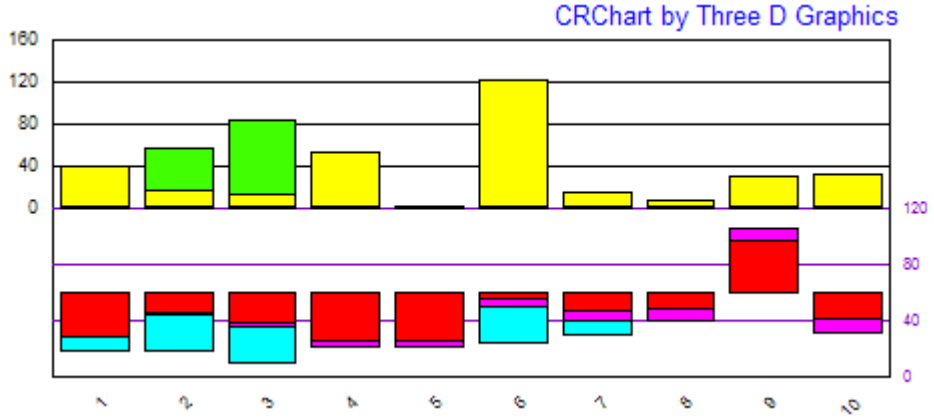
```
@Y2BASE f1
```

Parameters:

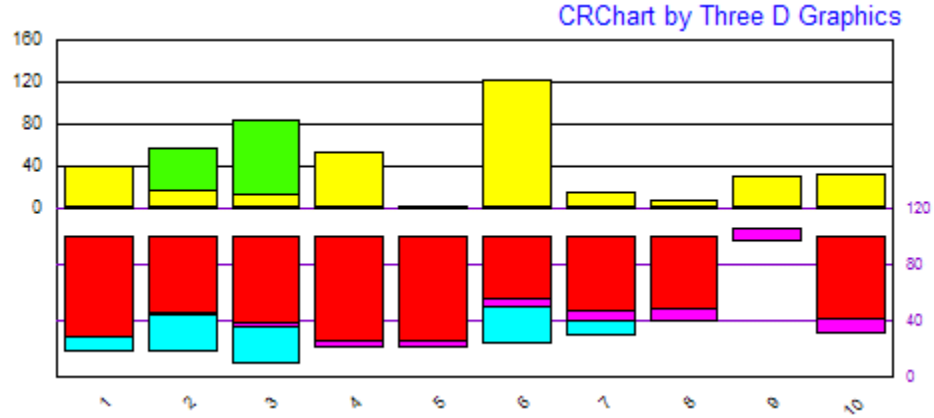
f1; Base line value

Example:

```
@Y2BASE 60
```




```
@Y2BASE 100
```



Persistent:

No

@Y2SLAVE (Slave Y2 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:

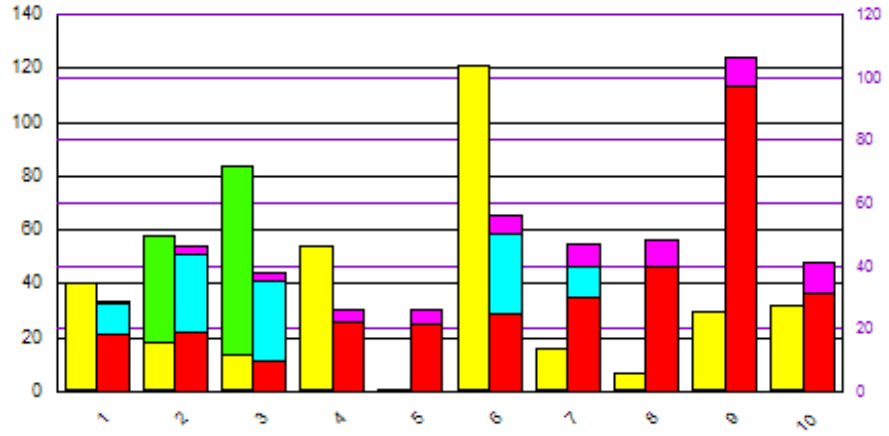
```
@Y2SLAVE
```

Parameters:

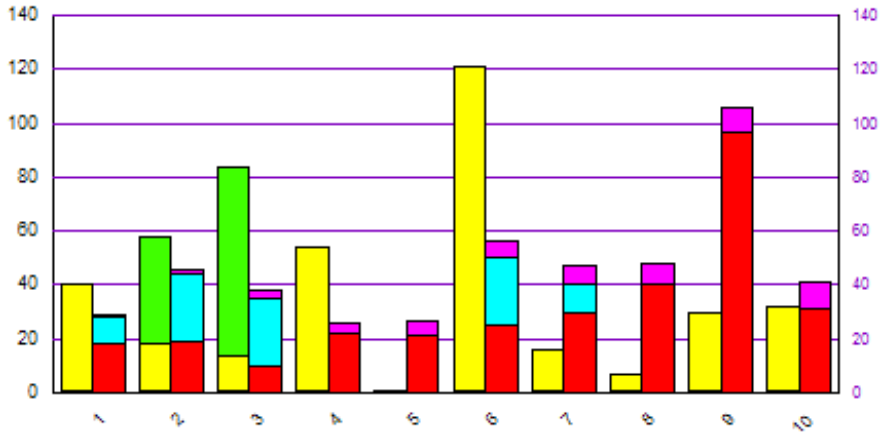
None

Example:

Before @Y2SLAVE



After @Y2SLAVE



Persistent:

Yes

@YSZ (Add Labels to Y-Lines)

This macro adds a user-defined line on the y-axis at value *fValue* with the label *sZLabel*.

Syntax:

```
@YSZ fValue nXFixup nYFixup sZLabel
```

Parameters:

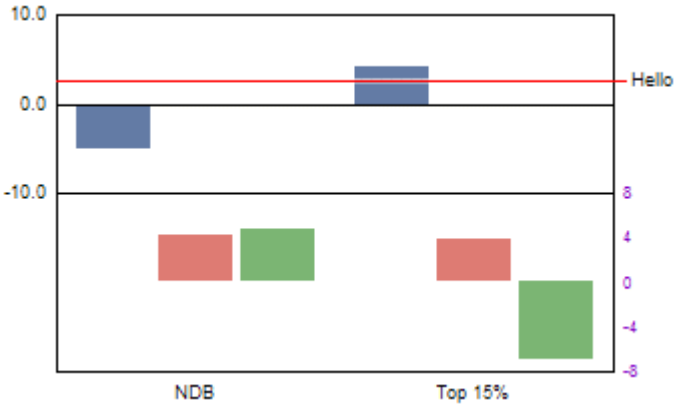
fValue; Value at which to add the user-defined line on the y-axis

nXFixup, nYFixup; Used to adjust the label to make closer to or further away from the line.

sZLabel; Label string to show next to line

Example:

```
@YSZ 3.2 0 0 Hello
```



Persistent:

No