



(Chart Enhancement Macros for Crystal Reports)

Version: 3.1

October, 2007

Three |D| Graphics | 11340 West Olympic Blvd., #352 | Los Angeles, CA. 90064
Telephone: 1.310.231.3330 | Fax: 1.310.231.3303 | Email: crchart@threedgraphics.com
Web: <http://www.threedgraphics.com>

Copyright:

Copyright 2003, 2004, 2005, 2006, 2007 by Three |D| Graphics. All rights reserved. This document may not be reproduced or disclosed in whole or in part by any means without the written consent of Three |D| Graphics.

Three |D| Graphics

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

Telephone: 1.310.231.3330

Fax: 1.310.231.3303

Email: crchart@threedgraphics.com

Web: <http://www.threedgraphics.com>

October, 2007

Table of Contents

Section 1: Overview	1-1
About CRChart	1-1
About CRChart Plus	1-1
System Requirements	1-1
Installation & Setup	1-2
Using The Installation Wizard	1-2
Deploying Your CRChart Solution (Manual Configuration)	1-3
Using CRChart with .NET on a Server	1-3
General Usage Notes	1-4
Reading CRChart Macros from the Registry	1-6
Order of Evaluation	1-6
Persistence	1-7
Compatibility with Existing Chart Library	1-8
Using CRChart Macros in Visual Basic	1-8
Error Reporting	1-8
Using Crystal Reports Fields/Functions in CRChart Macros	1-9
Getting Help	1-11
CRChart Viewer (CRViewer)	1-11
CRChart Plus-Specific Macros	1-14
Section 2: Auto Arrange Macros	2-1
@3DLABEL (3D Labels Fix up)	2-2
@AA (Auto Arrange)	2-3
@AA2 (Auto Arrange with Frame Adjustment)	2-4
@AA3 (Auto Arrange with Frame & Legend Adjustment)	2-5
@AA4 (Auto Arrange with Maximized Frame Adjustment)	2-6
@PARAM_FIXUP (Parameter Fix-Up)	2-7
Section 3: Axis Macros	3-1
@AXIS (Assign Series to Axis)	3-2
@DX (X-Axis Divisions)	3-3
@DY (Y-Axis Divisions)	3-4
@DY2 (Y2-Axis Divisions)	3-5
@GX (X-Axis Grid Style)	3-6
@GY (Y-Axis Grid Style)	3-7
@PX (X-Axis Precision)	3-8
@PY (Y1-Axis Precision)	3-9
@PY2 (Y2-Axis Precision)	3-10
@SC (Y-Axis Scale)	3-11
@SCX (X-Axis Scale)	3-12
@SCY2 (Y2-Axis Scale)	3-13
@TIMEAXIS (Time Axis)	3-14
@X_AXIS_MODE (X-Axis Mode)	3-15
@Y1_FORCE_PERCENT (Y1-Axis Percent Format)	3-16
@Y1_INVERT (Y1-Axis Invert)	3-17
@Y1BASE (Y1-Axis Base Line)	3-18
@Y2_FORCE_PERCENT (Y2-Axis Percent Format)	3-19
@Y2_INVERT (Y2-Axis Invert)	3-20
@Y2BASE (Y2-Axis Base Line)	3-21
@Y2SLAVE (Y2-Axis Slave to Y1)	3-22
@Y2SLAVE2 (Y1/Y2 Slave to Max Value)	3-23
Section 4: Series & Groups	4-1
@FORCE_ABSOLUTE (Force Series Absolute)	4-2

@FORCE_SERIES_COUNT (Force Series Count)	4-3
@FORECAST (Add Blank Groups)	4-4
@GM (Group Major)	4-5
@IG (Ignore Group)	4-6
@IS (Ignore Series)	4-7
@LIMIT_VISIBLE_GROUPS (Limit Visible Groups)	4-8
@PERCENT_SERIES (Create Ratio Series)	4-9
@RG (Reverse Group)	4-10
@RS (Reverse Series)	4-11
@SINGLE_GROUP (Single Group in Scatter Chart)	4-12
@SORT (Sort Series/Groups)	4-13
@STEP_LINE (Series Stepped Line)	4-16
@STEP_LINE2 (Series Stepped Line at Values)	4-17
@STOP (Force Assign Elements to Series One)	4-18
@SWAP (Swap Series/Groups)	4-19
@TOTAL_GROUP (Create a Total Group)	4-20
@USER_SERIES (User-Defined Series)	4-21
Section 5: Labels	5-1
@AGL (Alias Group Label)	5-2
@ASL (Alias Series Label)	5-3
@ASL_DP (Alias Series Label/Data Point)	5-4
@COND_GROUP_LABEL (Conditional Group Label)	5-5
@COND_GROUP_LABEL2 (Conditional Group Label 2)	5-6
@XSKIP (X-Axis Skip Labels)	5-7
@XSKIP2 (X-Axis Skip/Force Last Label)	5-8
Section 6: Riser & Markers	6-1
@HIDE_ZERO (Hide Zero Riser/Marker)	6-2
@HL (Highlight Riser/Marker)	6-3
@MARKER (Marker Shapes)	6-4
@MCOLOR (Marker Colors)	6-5
@PAT (Riser Pattern)	6-6
@RISER_BORDER (Riser Border)	6-7
@RISER_OVERLAP (Riser Overlap)	6-8
@RISER_WIDTH (Riser Width)	6-9
Section 7: Data & Data Text	7-1
@APPEND_DATATEXT (Append String to Data Text)	7-2
@DATATEXT (Data Text Mode)	7-3
@DLT (Data Line Type)	7-4
@DP (Data Point Override)	7-5
@DPC (Data Point Clear)	7-6
@DT (Show/Hide Series-Specific Data Text)	7-7
@NEG_STYLE (Negative Value Style)	7-8
@OFFSCALE_Y1 (Y1-Axis Offscale Values)	7-9
@RESET (Reset Internal Data Range)	7-10
@STRIP_ZERO (Strip Zero Values)	7-11
Section 8: Color & Special Effects	8-1
@ALPHA (Alpha Channel Transparency)	8-2
@BEVEL (Bevel Chart Object)	8-3
@COLOR_MODE (Color Mode)	8-5
@COND_COLOR & @COND_COLOR2 (Conditional Colors)	8-6
@CURVED_LINES (Enable Curved Lines)	8-8
@GCOLOR (Color Chart Object)	8-9

@HQ (Enable High-Quality Rendering)	8-10
@SHADOW (Drop Shadow)	8-11
Section 9: Lines	9-1
@CXY (X/Y Coordinates Line with Color)	9-2
@CY (Y-Axis Horizontal Line with Color)	9-3
@LS (Line Style)	9-4
@MEAN (Mean Line)	9-5
@MOVA (Moving Average Line)	9-6
@SMOOTH_LINE (Smooth/Straight Lines)	9-7
@TRENDLINE (Trend Line)	9-8
@TRENDLINE_ALLDATA (Linear Regression Line)	9-9
@X (X-Axis Vertical Line)	9-10
@XG (X-Axis Vertical Line at Group)	9-11
@XY (X/Y Coordinates Line)	9-12
@Y (Y1-Axis Horizontal Line)	9-13
@YSZ (Y-Axis Line with Label)	9-14
@YSZL (Y-Axis Line with Label on Left)	9-15
@YSZN (Y-Axis Line with Label & Value)	9-16
@YSZNL (Y-Axis Line with Label & Value on Left)	9-17
Section 10: User-Defined Objects	10-1
@USER_CIRCLE (Outlined Circle)	10-2
@USER_CIRCLE_ABOVE (Outlined Circle above Chart Area)	10-3
@USER_FILL (Color-Filled Rectangle)	10-4
@USER_FILL2 (Pattern-Filled Rectangle)	10-5
@USER_FILL_CIRCLE (Color-Filled Circle)	10-6
@USER_FILL_CIRCLE_ABOVE (Color-Filled Circle above Chart Area)	10-7
@USER_FILL_CIRCLE2 (Pattern-Filled Circle)	10-8
@USER_FILL_CIRCLE2_ABOVE (Pattern-Filled Circle above Chart Area)	10-9
@USER_MARKER (User-Defined Marker)	10-10
@USER_MARKER2 (User-Defined Marker with Value)	10-11
@USER_RECT (Outlined Rectangle)	10-12
Section 11: Chart Type & Chart Specific Macros	11-1
@3DSCAT (3D Scatter Chart)	11-2
@AUDIO (Audiogram)	11-3
@COMBO (Combo Chart)	11-4
@COMPARE2 (2-Series Absolute Bar Chart)	11-5
@GANTT (Gantt Chart)	11-6
@GRAPHTYPE (Graph Type)	11-7
@PARETO (Pareto Chart)	11-10
@POLAR (Polar Chart)	11-11
11.1 Macros for Box Plots	11-12
@BP/BP1 (Box Plot with Square Tail)	11-13
@BP2 (Box Plot with T-Style Tail)	11-14
@BP3 (Box Plot with I-Style Tail)	11-15
@BPH (Horizontal Box Plot)	11-16
@BPW (Box Plot Width)	11-17
@IN (Move First Box Plot In)	11-18
@MC (Marker Colors for Box Plots)	11-19
@MK (Number of Markers)	11-20
@MS (Marker Shapes for Box Plots)	11-21
@SZ (Size of Markers)	11-22
11.2 Macros for Gauges	11-23
@GAUGE_BORDER_STYLE (Gauge Border Style)	11-24

@GAUGE_BORDER_THICKNESS (Gauge Border Thickness)	11-25
@GAUGE_COLOR (Color Gauge Bands)	11-26
@GAUGE_MULTIPLE_NEEDLES (Multiple Needles)	11-27
@GAUGE_NEEDLE_STYLE (Gauge Needle Style)	11-28
@GAUGE_STYLE (Gauge Style)	11-30
@GAUGE_THRESHOLD (Gauge Band Threshold)	11-31
11.3 Macros for Pie Charts.....	11-32
@DATATEXT_PIE (Data Text Mode for Pie Charts).....	11-33
@PIE_NEG (Negative Values in Pie Legend)	11-34
@SMART_PIE_LABELS (Smart Pie Labels)	11-35
11.4 Macros for Waterfall Charts	11-36
@WATERFALL (Waterfall Chart).....	11-37
@WATERFALL2 (Waterfall Chart with Total Group)	11-38
@WF_CONNECT (Waterfall Feeler Lines).....	11-39
Section 12: Miscellaneous	12-1
@DEBUG (Show Debug Information).....	12-2
@EB (Error Bars)	12-3
@FONTANGLE (Font Angle)	12-4
@FONTSIZE (Font Size)	12-5
@HAT (Hat on Error Bars)	12-6
@LEGEND_ORDER (Force Legend Order)	12-7

Section 1: Overview

CRChart is a replacement charting library for Crystal Reports and Business Object XI. It adds significant new charting capabilities for Crystal Reports designers and developers. This “replacement library” is 100% compatible with your Crystal Reports and Business Object XI product. It simply expands your charting and graphing capabilities. CRChart is available in two configurations:

- CRChart Standard
- CRChart PLUS

About CRChart

The standard CRChart library includes features such as the following:

- 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.

About CRChart Plus

The CRChart Plus library includes all standard CRChart macros plus the following additional features:

- High-Quality Rendering
- Expanded Gauge Functionality
- Advanced Pie Label Layout Engine

System Requirements

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

- Microsoft Windows XP, NT, 2000 or Vista
- One of the following Crystal Reports:
 - Business Object XI Release 1 (Crystal Reports 11)
 - Business Object XI Release 2 (Crystal Reports 11.5)
 - Crystal Reports 10
 - Crystal Reports 9
 - Crystal Reports 8.5

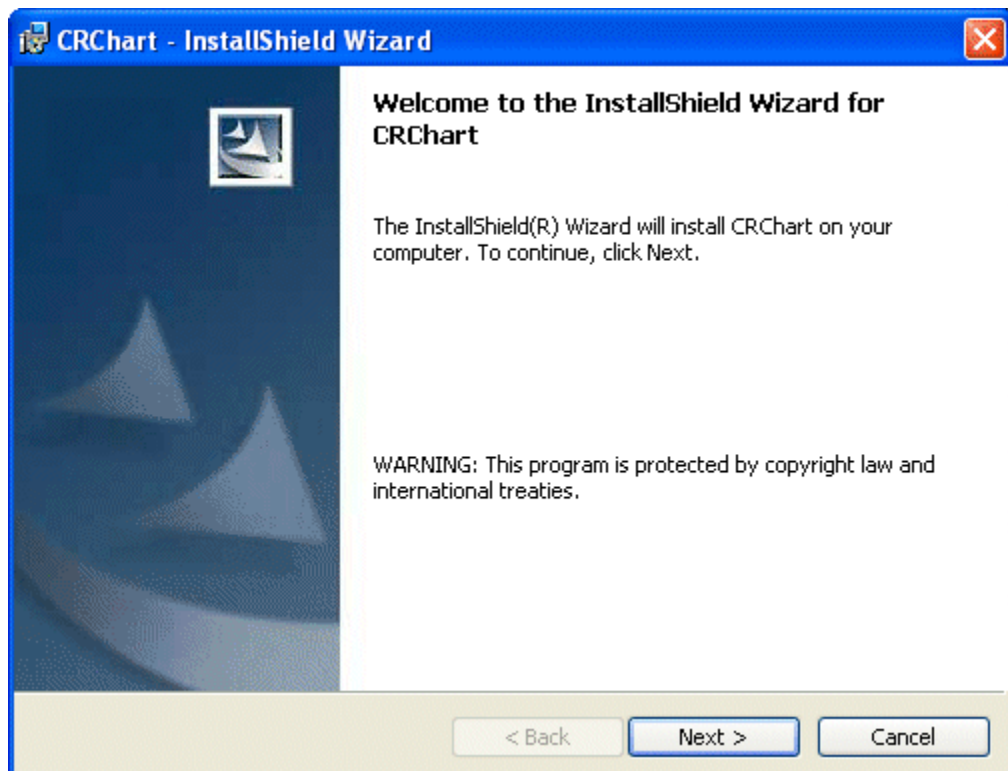
NOTES:

- Limited versions of Crystal Reports are included in some versions of Microsoft Visual Basic 6 and Visual Studio.NET. These stripped down version are not compatible with CRChart. You must have a real copy of Crystal Reports, such as Crystal Reports Standard or Developer Edition.
- Crystal Reports 8.5 is now only being supported for maintenance and bug fixes. As a result, all new CRChart macros (except @RISER_OVERLAP) in this and any subsequent versions of CRChart will not work with Crytsal Reports 8.5. See "New Macros in this Release of CRChart" for a list of new macros that are included in this version of CRChart.

Installation & Setup

Using The Installation Wizard

The CRChart Installation Wizard will guide you through the process of installing the library file in the appropriate location:



During this process, you will be asked to supply your CRChart serial number and the version of Crystal Reports you intend to use with CRChart. The Installation Wizard installs the latest documentation, sample code and reports, and the full libraries for all versions of Crystal Reports at the following location:

c:\Program Files\Three D Graphics\CRChart

A shortcut to the documentation is added to the Start Menu, under Three D Graphics -> CRChart.

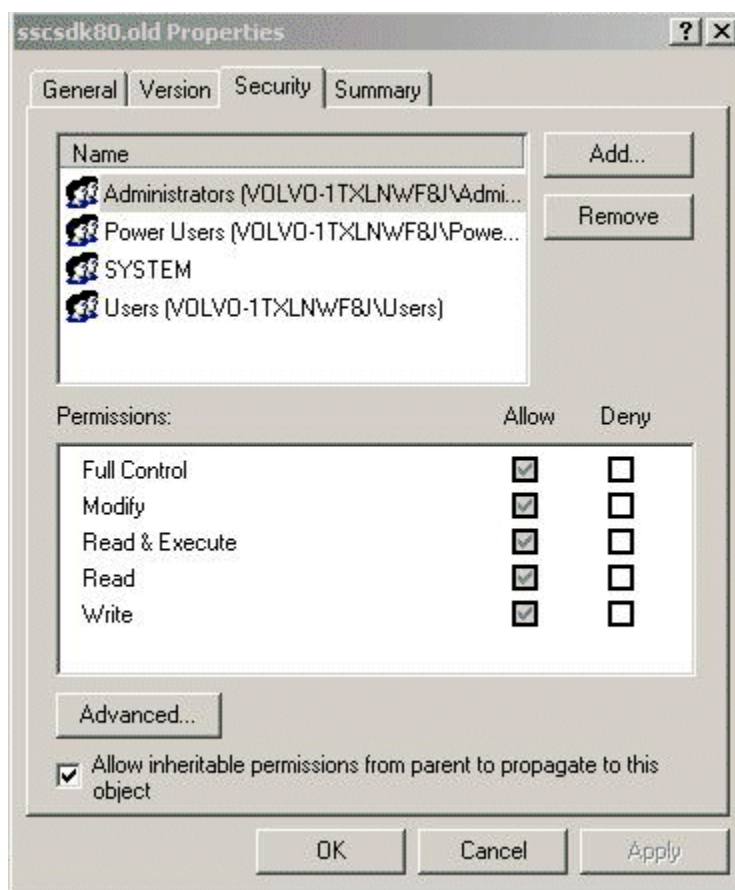
Deploying Your CRChart Solution (Manual Configuration)

For developers who are preparing applications for distribution that use CRChart macros, the charting library (sscsdk80.dll) file must be shipped with your application. Depending on the version of Crystal Reports you are using, the CRChart Installation Wizard will install the sscsdk80.dll file in one of the following locations:

Version	Path
11.5 (Business Object XI, Release 2)	C:\Program Files\Business Objects\Common\3.5\bin
11 (Business Object XI, Release 1)	C:\Program Files\Common Files\Business Objects\3.0\bin
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

Using CRChart with .NET on a Server

If CRChart is installed with Crystal Reports 10.x or 11.x (or higher) and .NET running on a server, you must set the security properties of sscsdk80.dll.

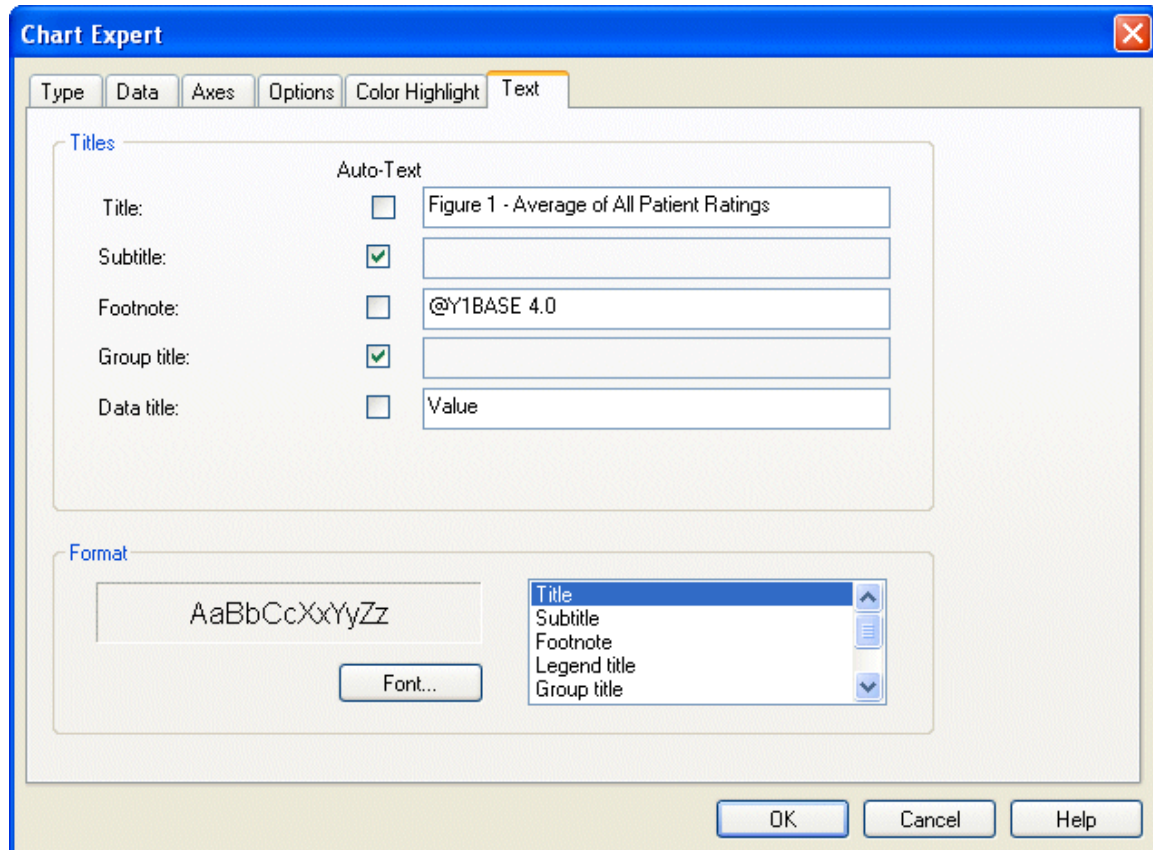


- Check the "Allow inheritable permission from parent to propagate to this object" checkbox.
- Click the OK button
- Restart the .NET server.

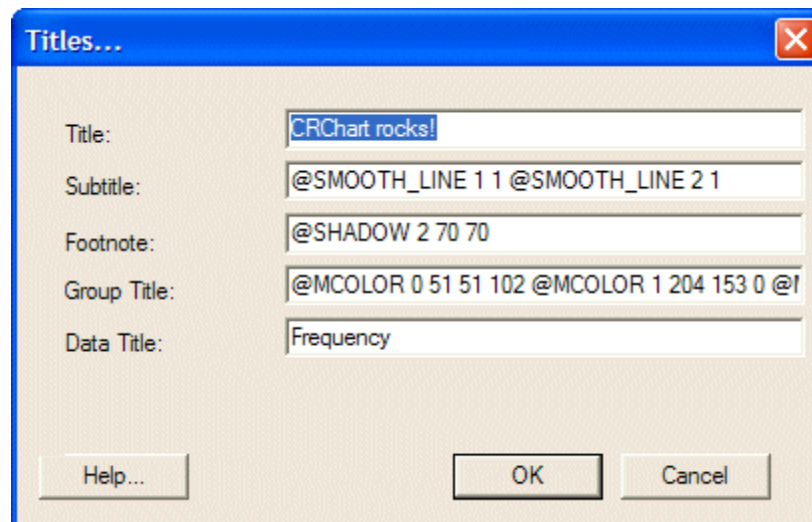
General Usage Notes

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.

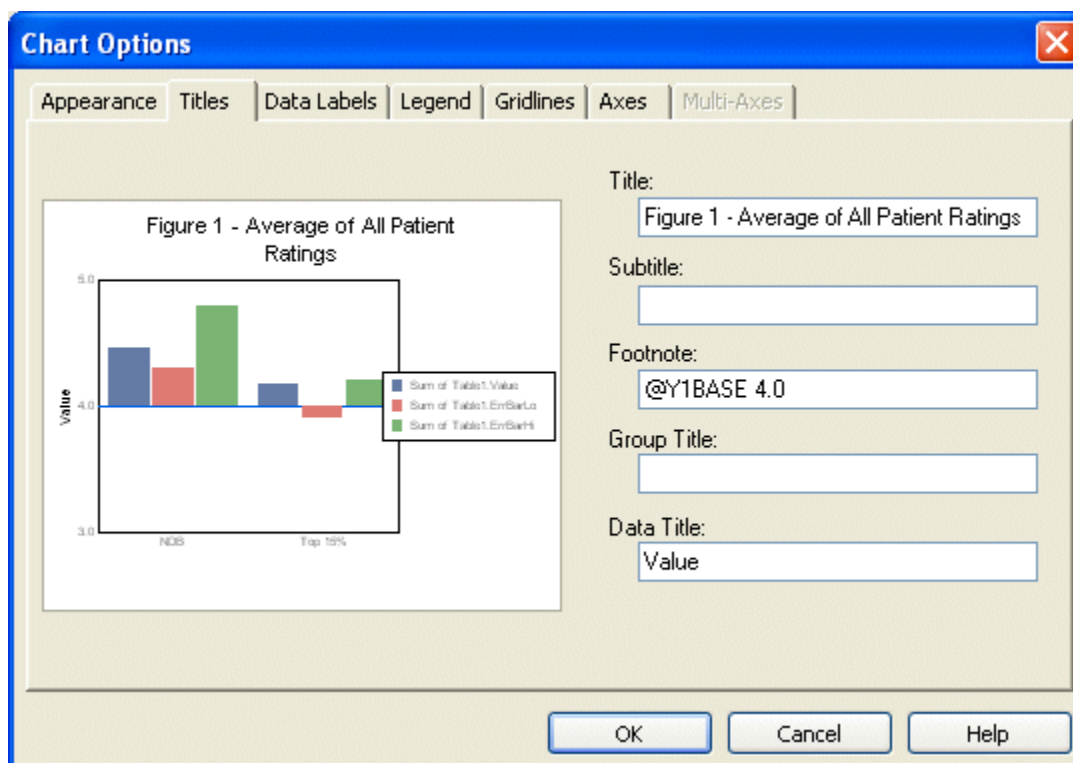
- Right-click on the chart background and choose Chart Expert to show the Chart Expert dialog. Click the Text tab to show chart title fields.



- For Crystal Reports Version 10 and lower, 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.



- For Crystal Reports Version 11 and higher, right click on the chart background and choose Chart Options to show the Chart Options dialog. Click the Titles tab to show chart title fields.



Even though macros are defined in the chart title fields, they will not appear as text in your chart. If you need to use a particular title field to define a chart title and to enter a macro, append a tilde (~) character and a space to the title and add the macro after the space. The space after the tilde is important. If it is not included, the macro will be ignored.

EXAMPLE:

```
TitleText~ @3DSCAT
```

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 MyNewSeries
```

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

EXAMPLE:

```
@BP2 @MK 8
```

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

EXAMPLE:

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

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

Reading CRChart Macros from the Registry

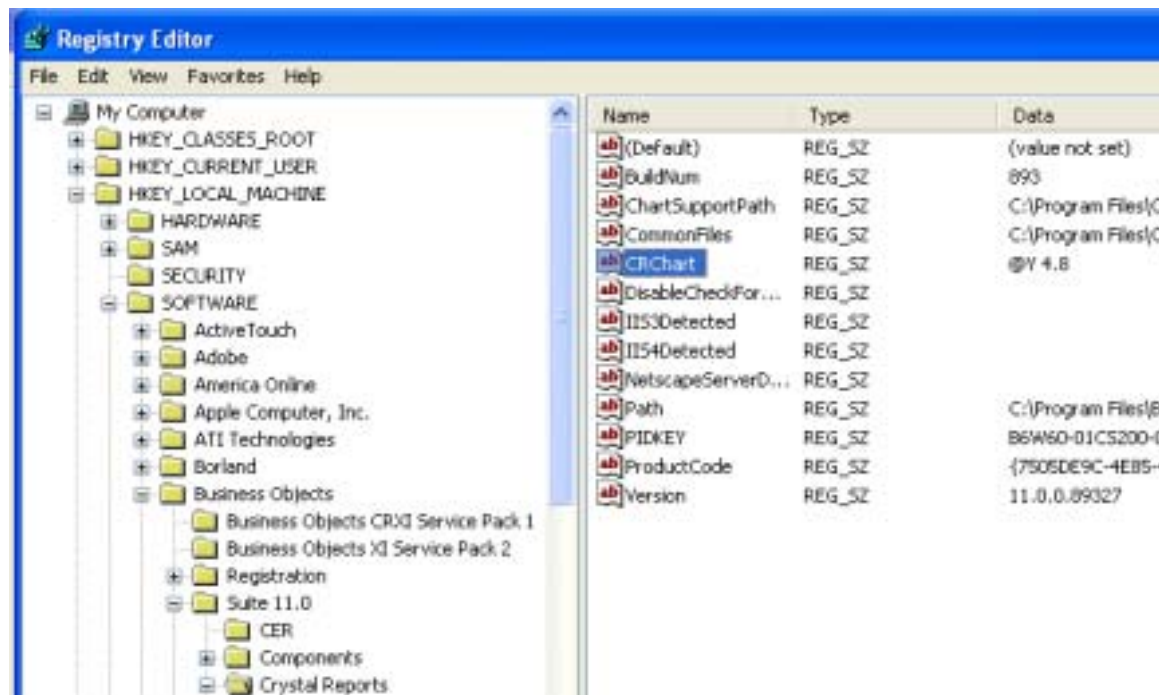
For Crystal Reports 11 and higher, CRChart can read macros directly from the registry. For Crystal Reports 11, CRChart will look for a CRChart key at:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Business Objects\Suite11.0\Crystal Reports
```

For Crystal Reports 11.5, CRChart will look for a CRChart key at:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Business Objects\Suite11.5\Crystal Reports
```

If CRChart finds the key at either of these locations, it will parse the string for macro commands in the key. Parsing occurs AFTER all other title strings are parsed. It is the last to be evaluated after of title, subtitle, footnote, Y1-title, X-title, etc.

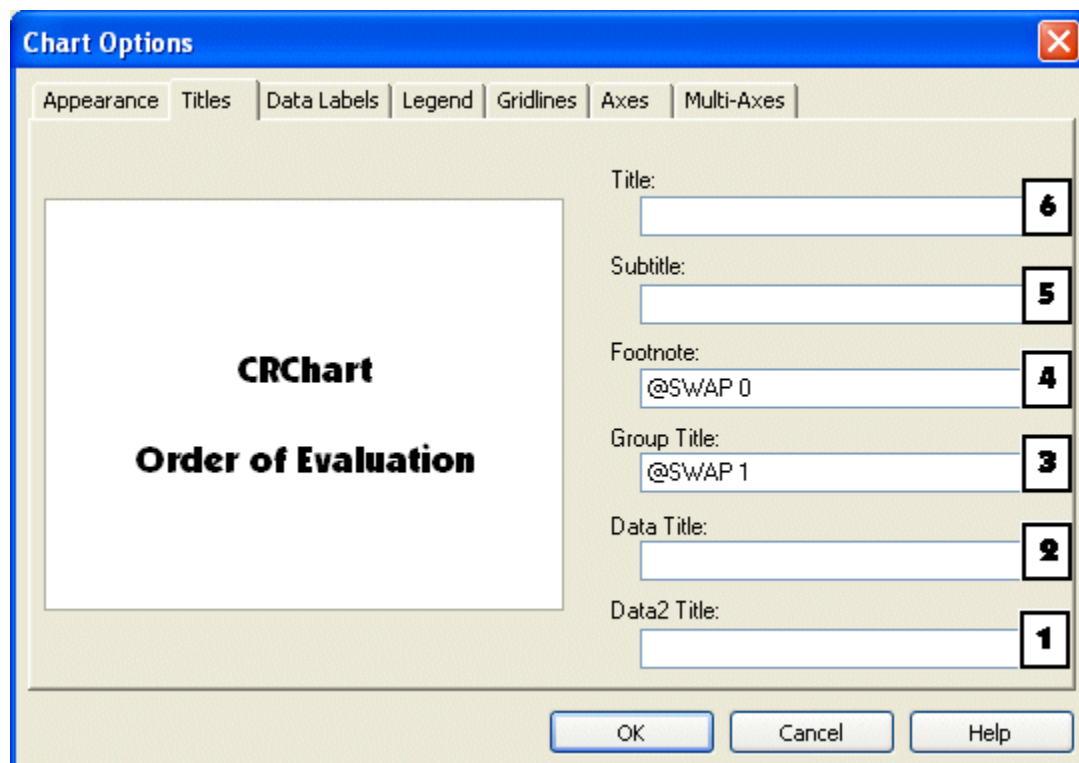
EXAMPLE:**Order of Evaluation**

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) or in a registry key, you could potentially define conflicting macros. To

avoid this potential conflict, the macros are evaluated in the following order:

1. Data2 (Y2-Axis) Title
2. Data (Y1-Axis) Title
3. Group (X-Axis) Title
4. Footnote
5. Subtitle
6. Title
7. Macro defined in registry key (Crystal Reports 11 or higher only)

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



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

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

@SWAP 0 @SWAP 1

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

Persistence

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

macro is removed. A "@SWAP 0" macro is needed to return the series/groups to their normal/default order.

Some of the chart enhancement macros are NOT persistent and their effect will disappear when the macro is removed. For example the user-defined lines (set by @X, @Y, 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. We work closely with Business Objects to ensure that CRChart is continuously upgraded to exactly match any bug fixes, enhancements or service pack changes made by Business Objects.

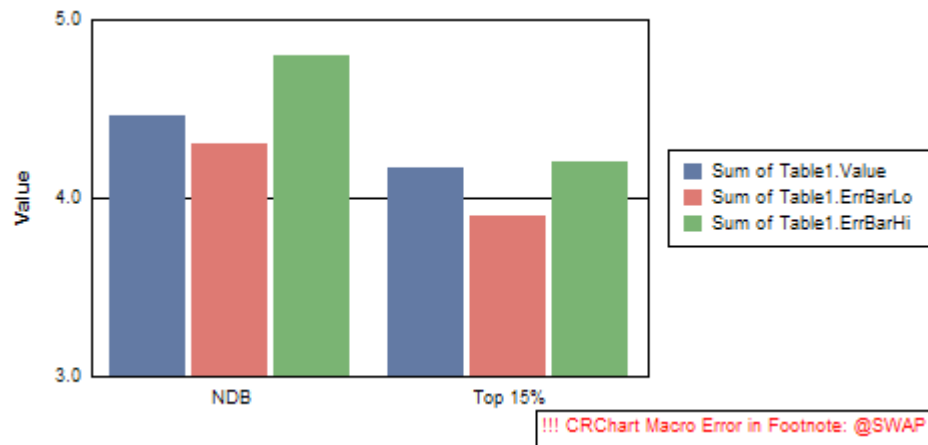
Using CRChart Macros in Visual Basic

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
```

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 Crystal Reports Fields/Functions in CRChart Macros

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

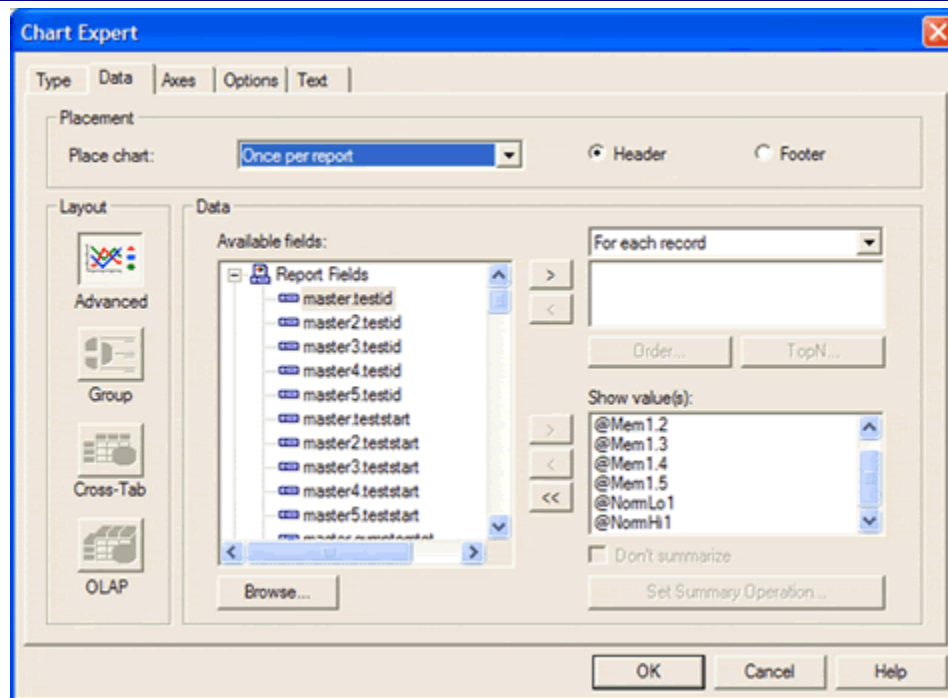
- Add the report field or function that you want to use with a CRChart macro to the end of "Show value(s)" list in the Data tab of the Chart Expert dialog.
- Make a note of the zero-based position of this field in the "Show value(s)" list.
- In your CRChart macro, use the letter "P" followed by the number 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).
- Currently, only NUMERIC parameters can be mapped (integers, Booleans, reals). CRChart does not currently support runtime mapping of string parameters.

EXAMPLE:

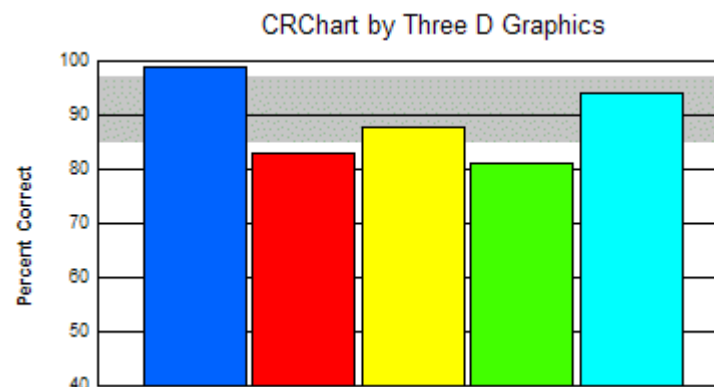
In the Data tab of the Chart Expert dialog, the "NormLo1" and "NormHi1" fields are added at zero-based positions 5 and 6 in the "Show value(s)" list.



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.



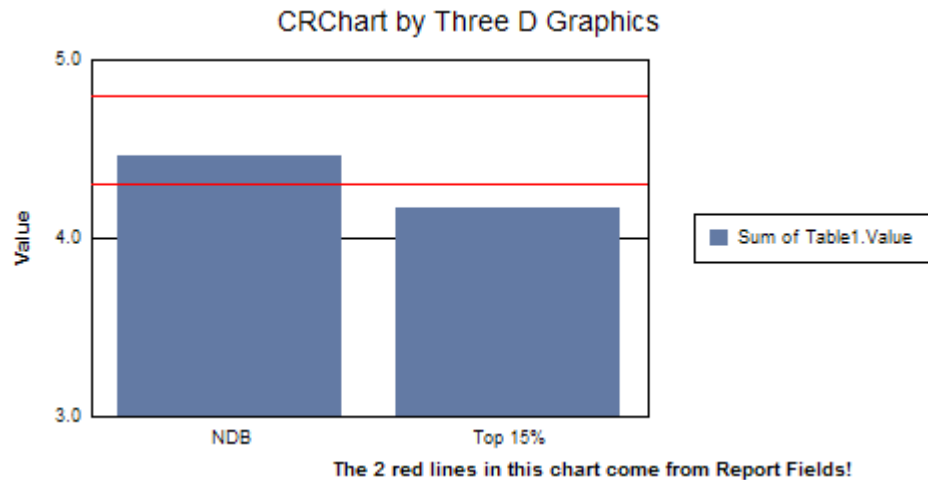
In the following example, the following fields are listed in the "Show value(s)" list in the Data Tab of Chart Expert:

```
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:



Getting Help

If you have any problems or questions about using CRChart, there are several resources from which to obtain additional information.

If you have a valid support contract, use one of the following:

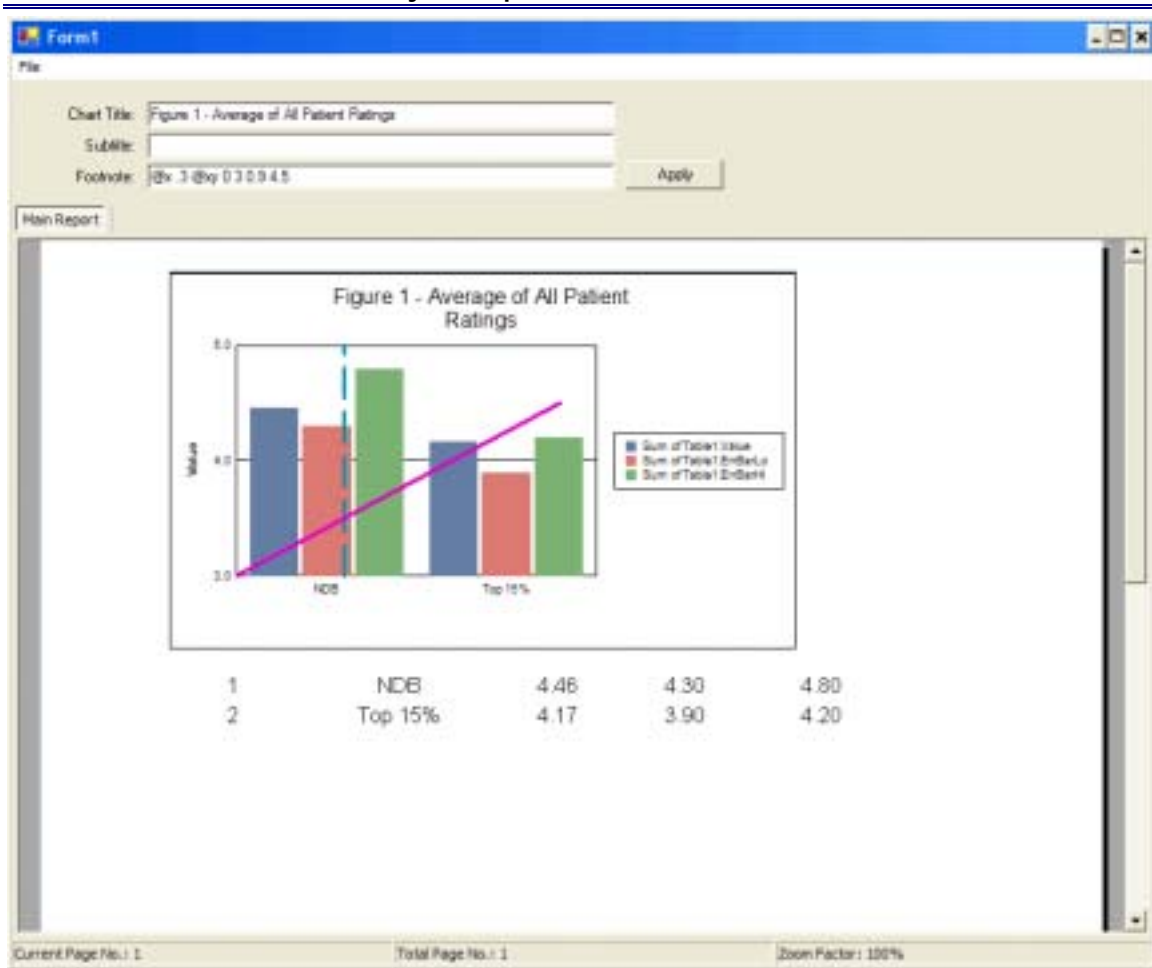
- Fill out the troubleshooting form at:
http://threedgraphics.com/tdg/products/tools/crchart/support_request.php
- Contact the CRChart support department: **e-Mail:**
crchart@threedgraphics.com or **Telephone:** (310) 231-3330

For CRChart Developers, use one of the following resources:

- The list of Frequently Asked Questions on the CRChart support page.
- Post your questions to fellow developers on the CRChart message board at: <http://www.threedgraphics.com/tdg/mb/>

CRChart Viewer (CRViewer)

The CRChart Viewer (CRViewer) allows you to load any Crystal Reports report file (.rpt) that includes chart and apply CRChart macros. CRViewer allows you to examine the results of CRChart macros outside of the Crystal Reports application. Example:



NOTE: CRViewer requires Crystal Report Application Server. Report Application Server (RAS) is included in Crystal Reports Enterprise 10 and 11. See the CRViewer Usage Notes for more details.

When you launch the CRViewer application, it will attempt to load the report named "UserLineExample_After.rpt" from the Crystal Reports Application Server Reports directory (by default C:\Program Files\Crystal Decisions\Report Application Server 10\Reports). If the UserLineExample_After.rpt file does not exist at this location, use the File menu Load Report option to load a .rpt file into the viewer. Enter CRChart macros or text in the Chart Title, Subtitle, and Footnote fields. As shown in this example, the chart title "Figure 1 - Average of All Patient Ratings" is entered in the Chart Title field. CRChart macros @X .3 @XY 0.3 0.9 4.5 are entered in the Footnote field. Click the Apply button to apply the text and/or CRChart macros to the chart and see the result.

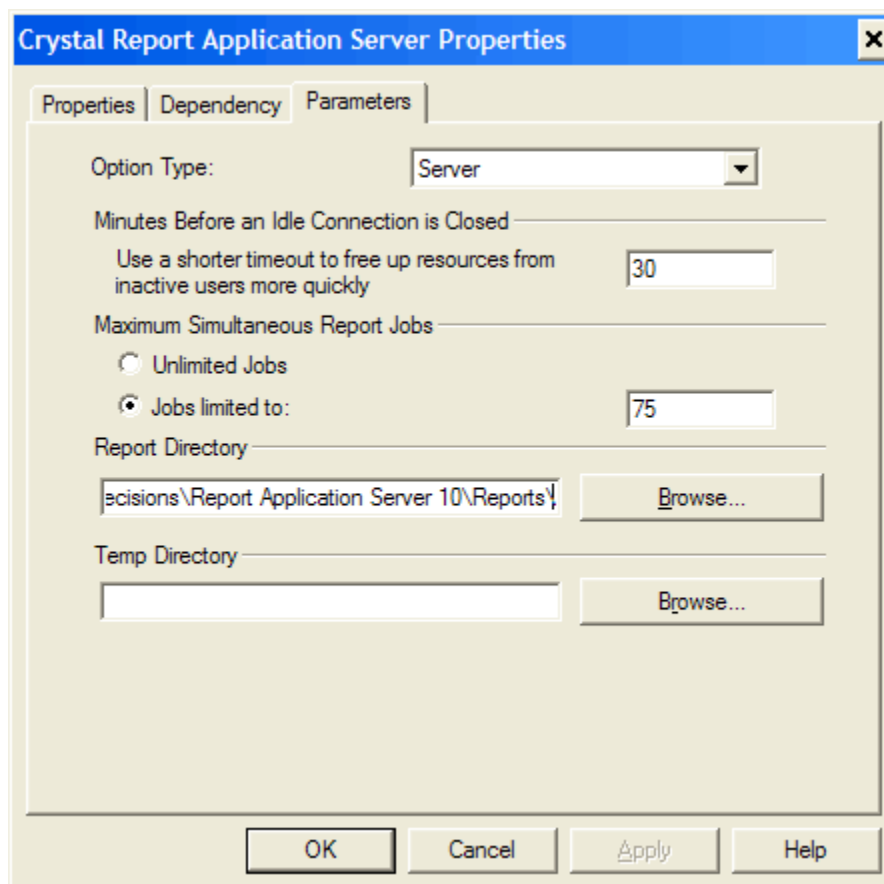
CRViewer is not included in the CRChart installation package. It can be downloaded separately from one of the following locations:

http://threedgraphics.com/tdg/products/tools/crchart/download_demo.php
<http://threedgraphics.com/tdg/products/tools/crchart/binaries/crviewer.zip>

The .ZIP file contains the CRViewer executable application (CRViewer.exe), source code, and example reports.

CRViewer Usage Notes:

- CRViewer requires Crystal Report Application Server. Report Application Server (RAS) is included in Crystal Reports Enterprise 10 and 11.
- If RAS is installed and running at the time you install CRViewer, you must stop and restart RAS before using CRViewer. (i.e., Select Start - > Crystal Enterprise 10-> Crystal Configuration Manager. Choose Crystal Report Application Server in the toolbar, click Stop, then Start.)
- CRViewer can only open reports from directories that are readable by RAS. Typically, this is the Report directory in the RAS install directory (c:\Program Files\Crystal Decisions\Report Application Server 10\Reports for CR 10). You will need to copy the sample .rpt files to that directory. The Parameters tab in the RAS Properties dialog shows the directory that is readable by RAS. Example:



- The CRViewer .ZIP file also includes the source code for the CRViewer application. You can load the build (CRViewer.sln) in Visual Studio 2003, modify it as required, and re-build the application to suit your needs. It is a straightforward C# application.

CRChart Plus-Specific Macros

The following macros have been added to this document to support this release of CRChart. All of these macros require the CRChart Plus charting library:

- @BEVEL; Draw a bevel effect on a chart object
- @CURVED_LINES; Enable/Disable curved lines in a line or area chart
- @GAUGE_BORDER_STYLE; Apply a gauge border style
- @GAUGE_BORDER_THICKNESS; Define the thickness of the gauge border
- @GAUGE_COLOR; Define the color of a gauge band
- @GAUGE_MULTIPLE_NEEDLES; Draw only one or all series as gauge needles
- @GAUGE_NEEDLE_STYLE; Select a gauge needle style
- @GAUGE_STYLE; Enable/Disable enhanced gauge drawing engine.
- @HQ; Enable/Disable High Quality rendering
- @SMART_PIE_LABELS; Enable/Disable enhanced pie label layout engine

Section 2: Auto Arrange Macros

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

- @3DLABEL; Automatically arrange labels in a 3D Chart.
- @AA; Automatically arrange objects in a chart.
- @AA2; Automatically arrange objects in a chart with Frame adjustment.
- @AA3; Automatically arrange objects in a chart with Frame & Legend adjustment.
- @AA4; Automatically arrange objects in a chart with Maximized Frame adjustment.
- @PARAM_FIXUP; Fix parameters from Crystal Reports Fields/Functions.

@3DLABEL (3D Labels Fix up)

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

SYNTAX:

```
@3DLABEL nID nX nY
```

PARAMETERS:

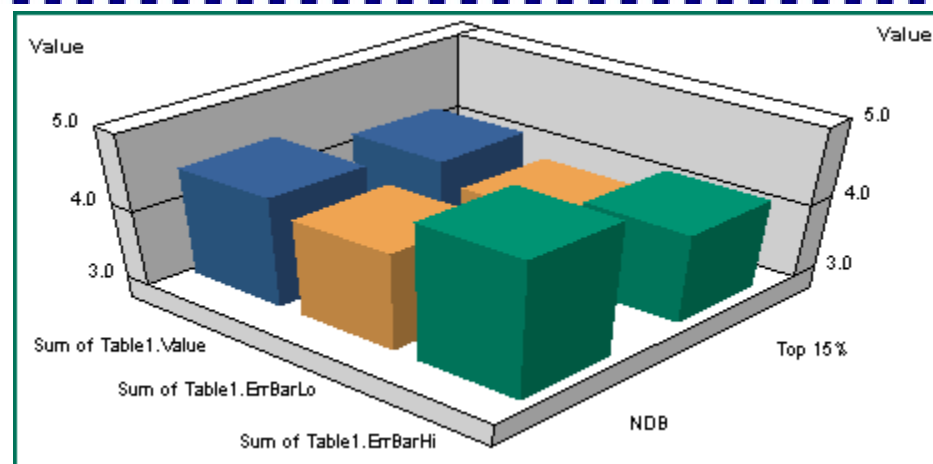
nID = 0...3 selects which labels to fix. 0=group labels (normally along the right floor of the 3D cube), 1=series labels (normally along the left floor of the 3D cube), 2=labels on the left side of the 3D cube, 3=labels on the right side of the 3D cube.

nX = -2000....2000 selects the amount of fix up to do in the X-direction

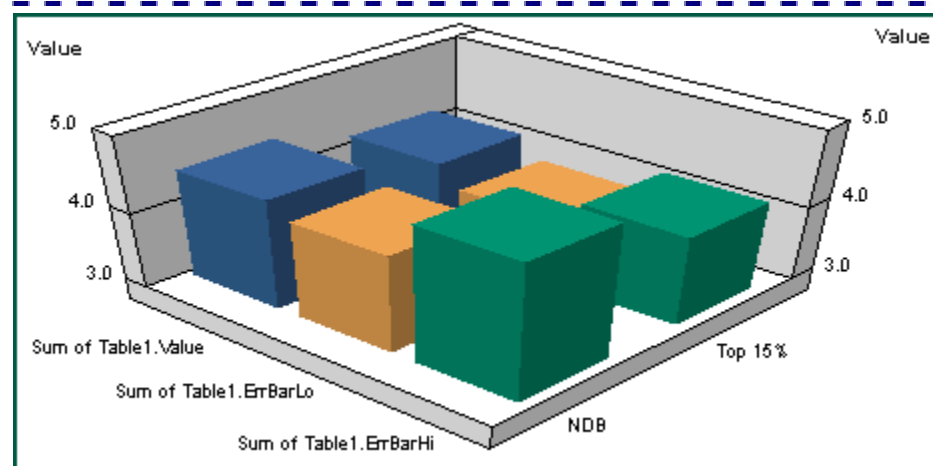
nY = -2000....2000 selects the amount of fix up to do in the Y-direction

EXAMPLE:

```
@3DLABEL 0 2000 0
```



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



PERSISTENT:

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.

@AA (Auto Arrange)

This macro automatically arranges elements in a chart.

SYNTAX:

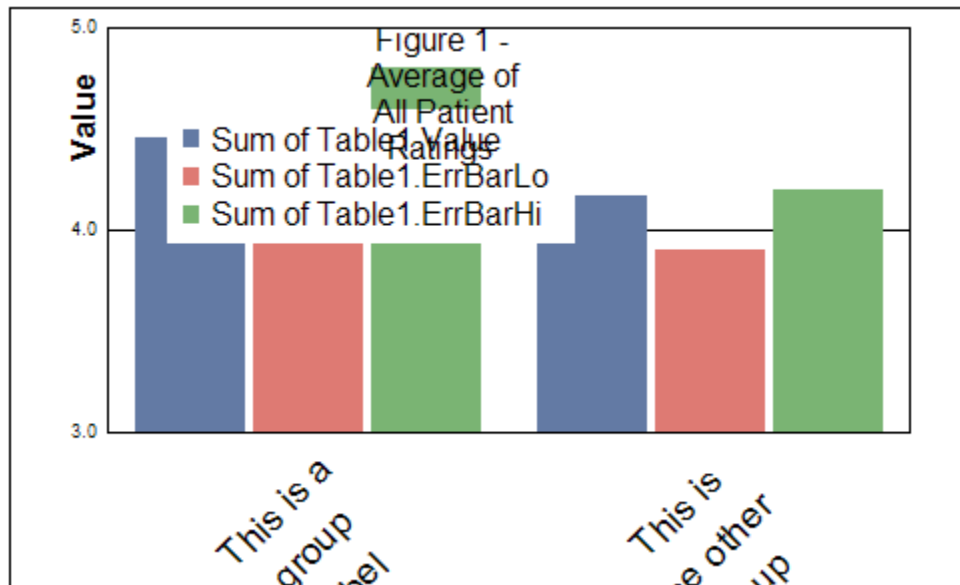
@AA

PARAMETERS:

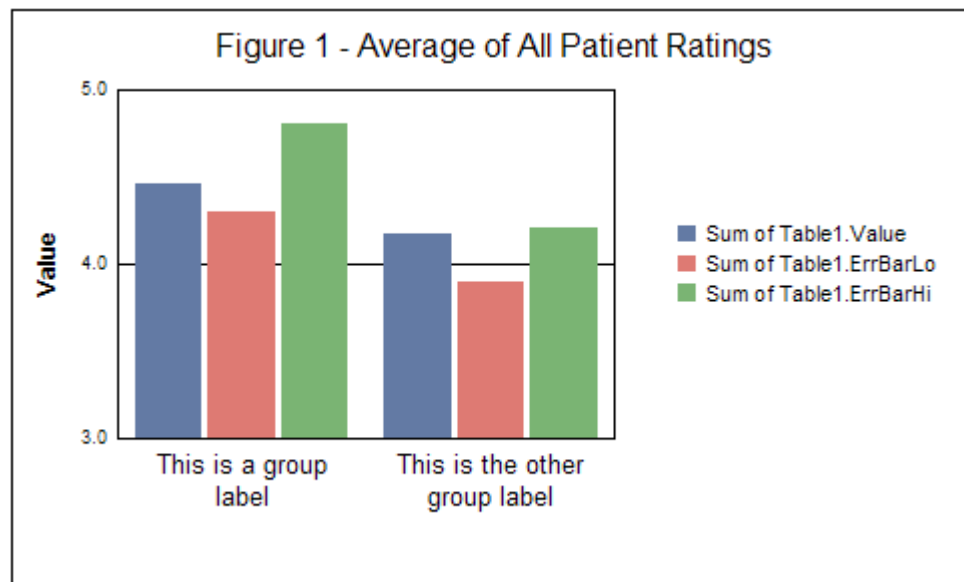
None

EXAMPLE:

Before @AA



After @AA macro applied



PERSISTENT:

YES

@AA2 (Auto Arrange with Frame Adjustment)

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

- 1) It does not autosize font labels. Each label keeps its current font size.
- 2) The *nFixup* parameter can be used to move the bottom of the chart frame up or down to provide more or less space for the X-axis labels

SYNTAX:

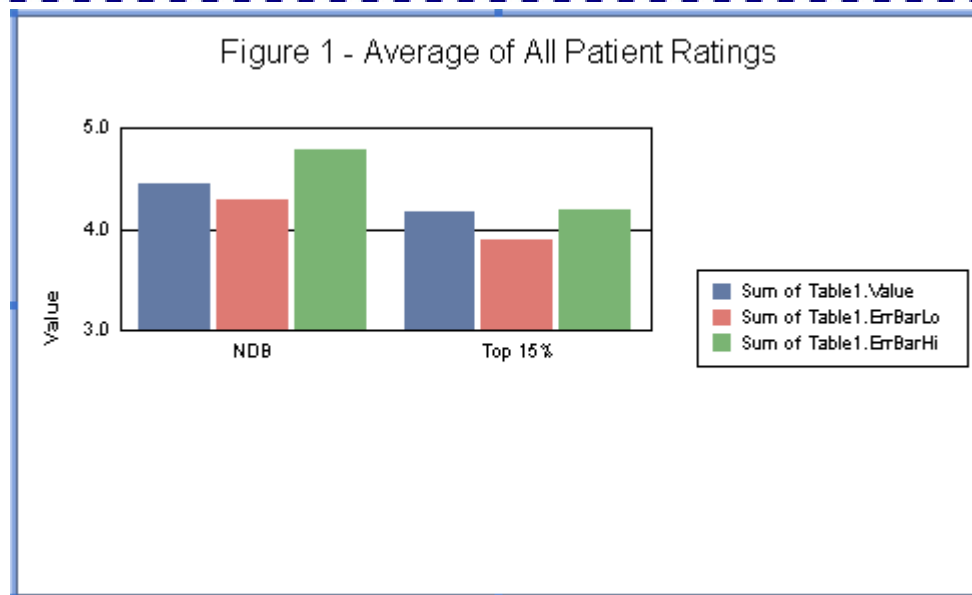
```
@AA2 nFixup
```

PARAMETERS:

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

EXAMPLE:

```
@AA2 -10000
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@AA3 (Auto Arrange with Frame & Legend Adjustment)

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

- 1) It does not autosize font labels. Each label keeps its current font size.
- 2) The *nFixup* parameter can be used to move the bottom of the chart frame up or down to provide more or less room space for the X-axis labels.
- 3) The *nLegendMode* parameter can be used to force the legend to a specific location.

SYNTAX:

```
@AA3 nFixup nLegendMode
```

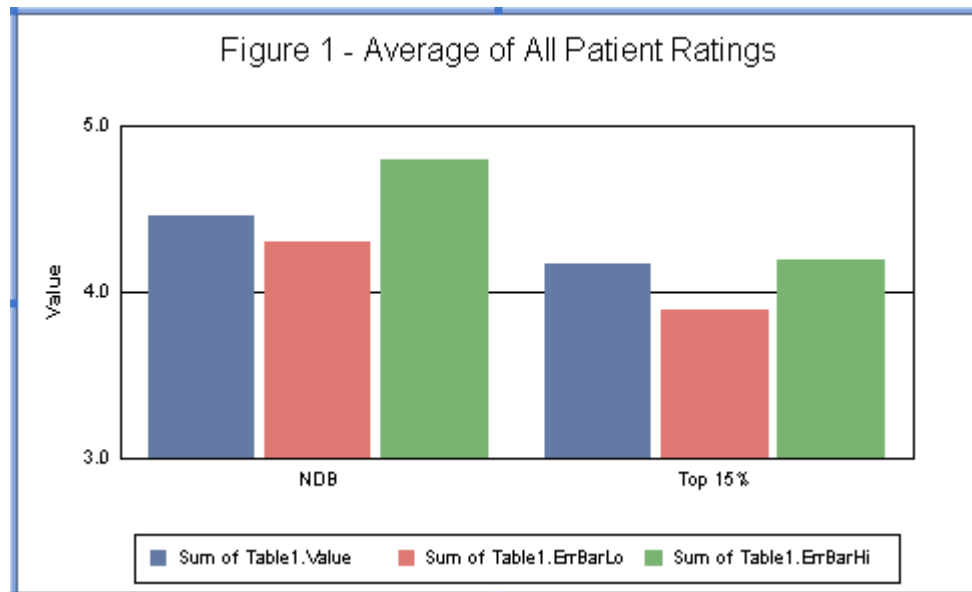
PARAMETERS:

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

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

EXAMPLE:

```
@AA3 -10000 2
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@AA4 (Auto Arrange with Maximized Frame Adjustment)

This macro is the same as @AA2 except it more accurately maximizes the chart frame with there are no titles. The *nFixup* parameter can be used to move the bottom of the chart frame up or down to provide more or less space for the X-axis/Group labels

SYNTAX:

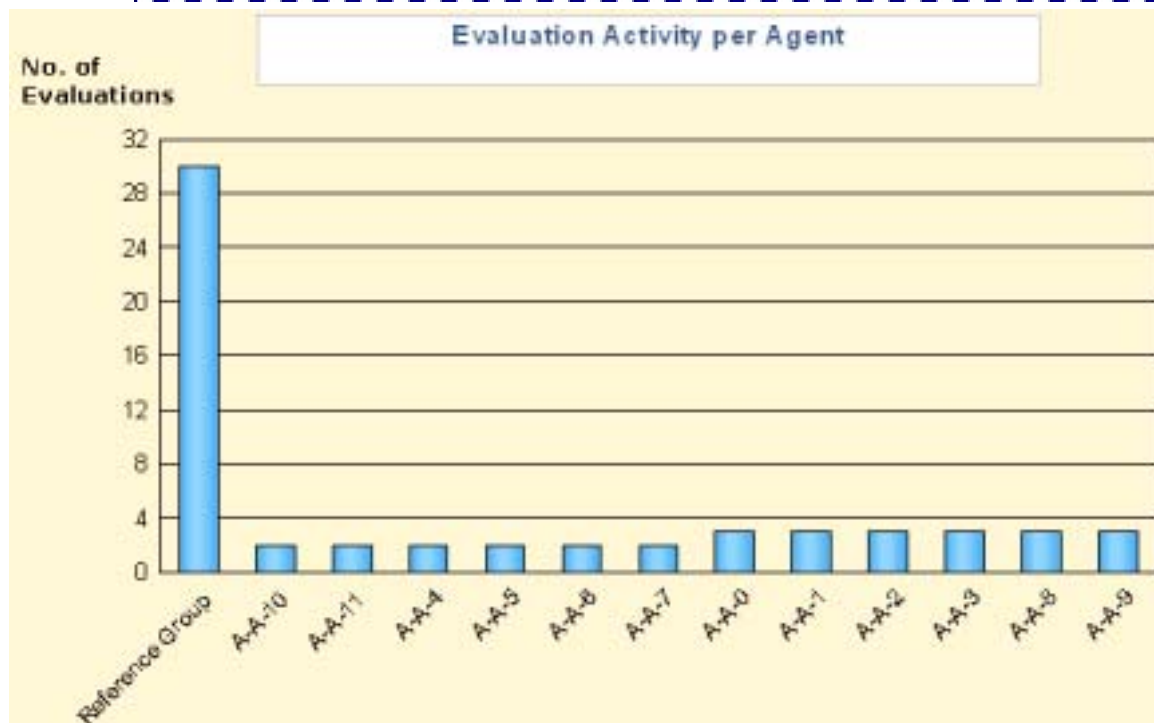
```
@AA4 nFixup
```

PARAMETERS:

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

EXAMPLE:

```
@AA4 0
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 10.x or higher.

@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 charts with a numeric X-axis.

SYNTAX:

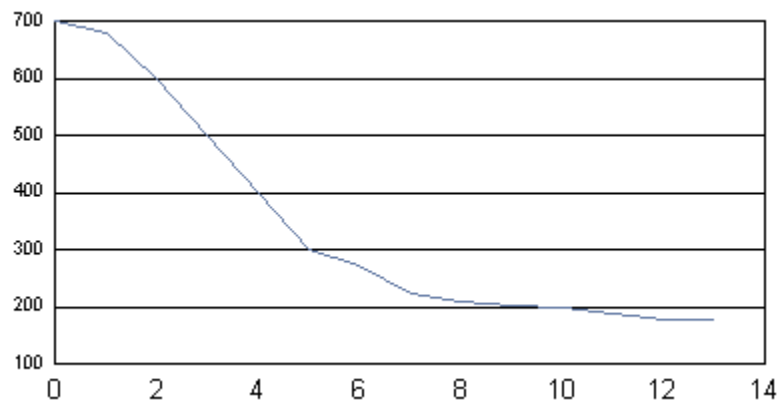
```
@PARAM_FIXUP nIndex
```

PARAMETERS:

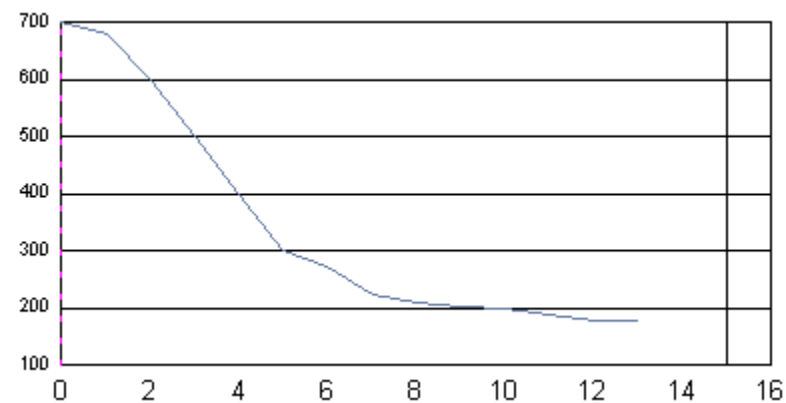
nIndex; 1=Enable parameter fix-up, 0=Disable parameter fix-up

EXAMPLE:

```
@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



Section 3: Axis Macros

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

- @AXIS; Assign a Series to an Axis
- @DX; Define the number of X-Axis Divisions
- @DY; Define the number of Y-Axis Divisions
- @DY2; Define the number of Y2-Axis Divisions
- @GX; Select a Grid Style for the X-Axis
- @GY; Select a Grid Style for the Y-Axis
- @PX; Define X-Axis Precision
- @PY; Define Y1-Axis Precision
- @PY2; Define Y2-Axis Precision
- @SC; Define Y-Axis Scale
- @SCX; Define X-Axis Scale
- @SCY2; Define Y2-Axis Scale
- @TIMEAXIS; Create Time Axis
- @X_AXIS_MODE; Define the X-Axis Mode
- @Y1_FORCE_PERCENT; For the Y1-Axis to use Percent Format
- @Y1_INVERT; Invert the Y1-Axis Invert
- @Y1BASE; Defined the Y1-Axis Base Line
- @Y2_FORCE_PERCENT; For the Y2-Axis to use Percent Format
- @Y2_INVERT; Invert the Y2-Axis Invert
- @Y2BASE; Defined the Y2-Axis Base Line
- @Y2SLAVE; Slave the Y2-Axis to the Y1-Axis
- @Y2SLAVE2; Slave the Y1/Y2-axes to Max Value

@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
```

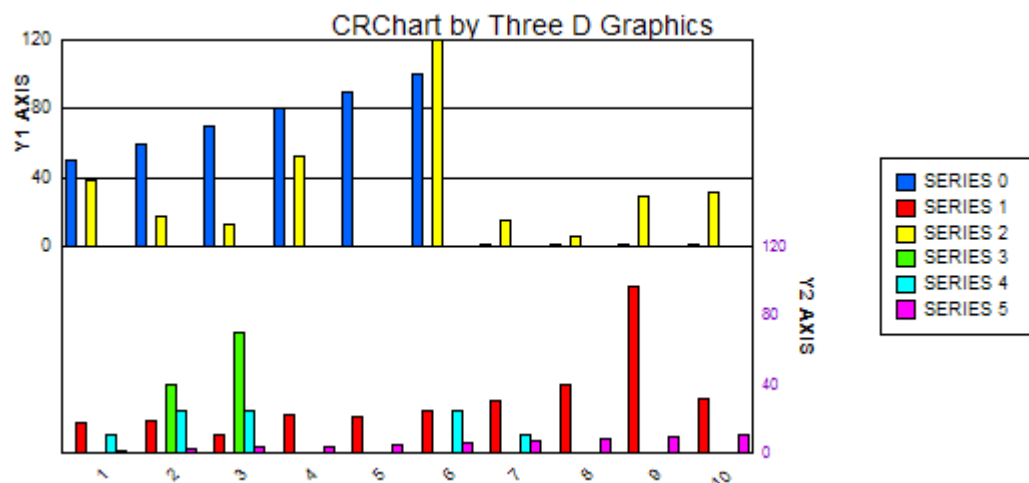
PARAMETERS:

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

@DX (X-Axis Divisions)

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

SYNTAX:

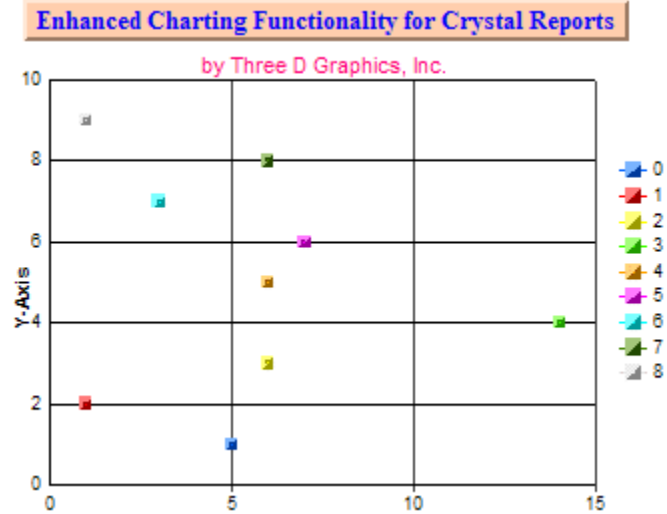
```
@DX n
```

PARAMETERS:

n; Number of divisions on the X-Axis

EXAMPLE:

```
@DX 2
```



PERSISTENT:

YES

ALSO SEE:

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

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

@DY (Y-Axis Divisions)

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

SYNTAX:

```
@DY n
```

PARAMETERS:

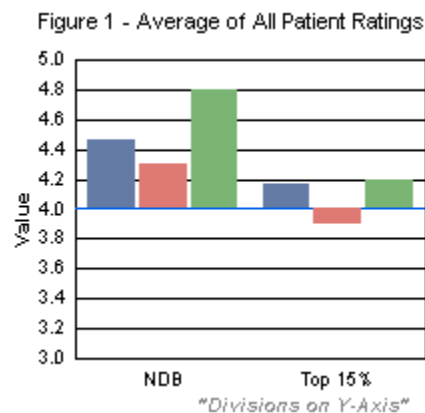
n; Number of divisions on the Y-Axis

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 to set the number of divisions on the X-Axis

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

@DY2 (Y2-Axis Divisions)

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

SYNTAX:

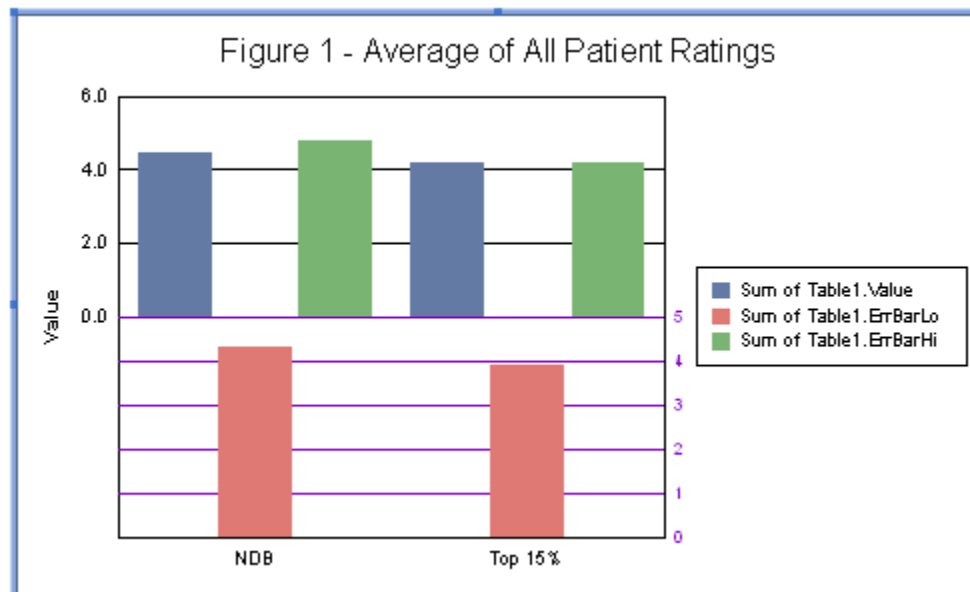
```
@DY2 n
```

PARAMETERS:

n; Number of divisions on the Y2-Axis

EXAMPLE:

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



PERSISTENT:

YES

NOTES:

This macro requires Crystal Report 9.x or higher.

ALSO SEE:

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

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

@GX (X-Axis Grid Style)

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.

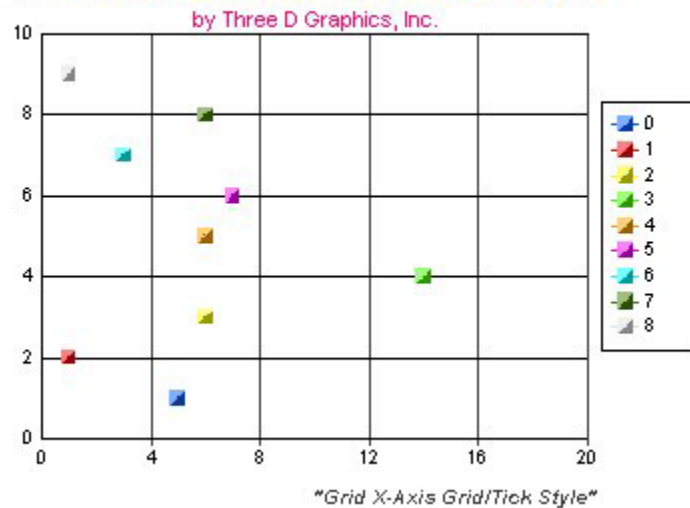
4 = No Grid. Outer Tick.

5 = No Grid. Inner and Outer Tick.

EXAMPLE:

@GX 2

Enhanced Charting Functionality for Crystal Reports



PERSISTENT:

NO

ALSO SEE:

@GY

@GY (Y-Axis Grid 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.

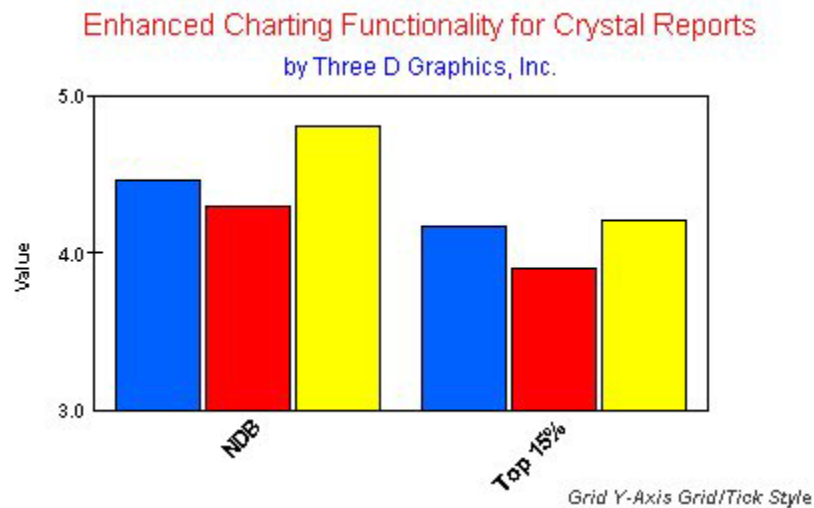
4 = No Grid. Outer Tick.

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

@PX (X-Axis Precision)

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

SYNTAX:

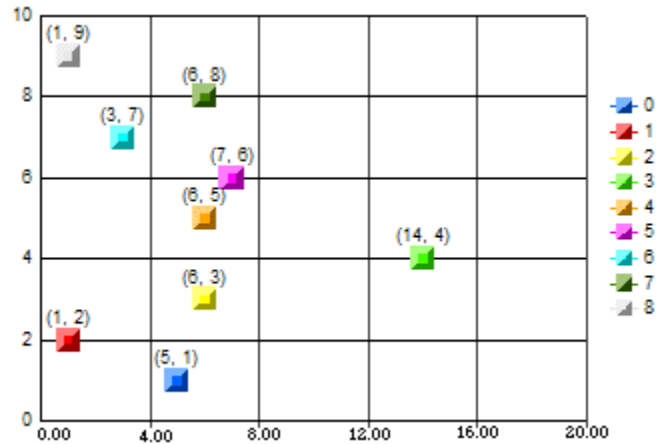
```
@PX n
```

PARAMETERS:

n; Number of decimal places to use

EXAMPLE:

```
@PX 2
```



In this example chart, X-axis values are shown on the bottom of the chart.

PERSISTENT:

YES

ALSO SEE:

@PY

@PY (Y1-Axis Precision)

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

SYNTAX:

```
@PY n
```

PARAMETERS:

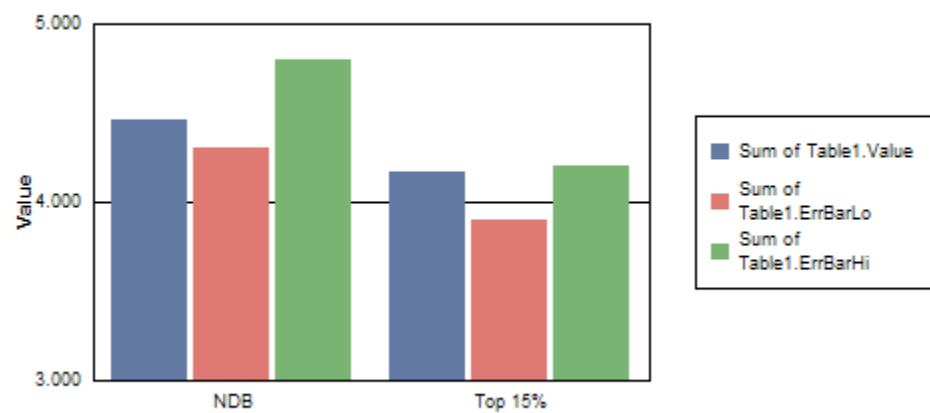
n; Number of decimal places to use

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



In this example chart, Y-axis values are shown on the left side of the chart.

PERSISTENT:

YES

ALSO SEE:

@PX, @PY2

@PY2 (Y2-Axis Precision)

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

SYNTAX:

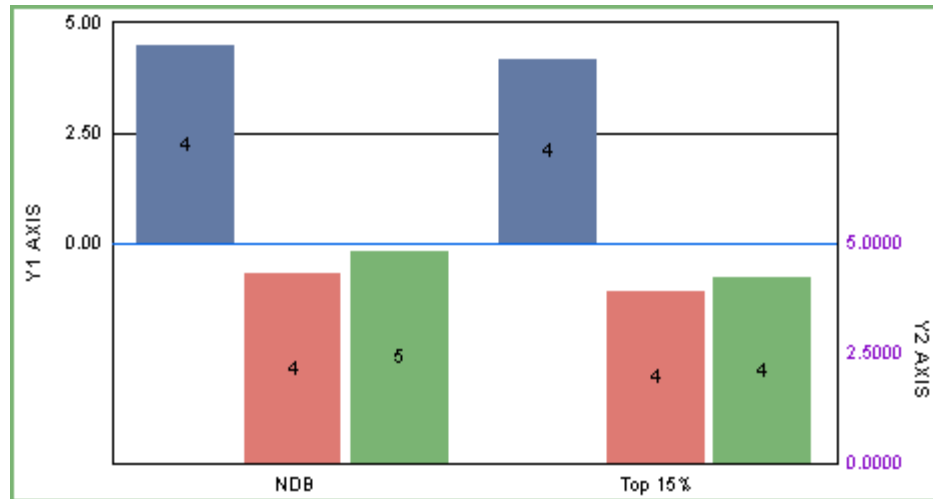
```
@PY2 n
```

PARAMETERS:

n; Number of decimal places to use

EXAMPLE:

```
@PY 2  
@PY2 4
```



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

PERSISTENT:

YES

ALSO SEE:

@PX, @PY

NOTES:

This macro requires Crystal Reports 9.x or higher.

@SC (Y-Axis Scale)

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

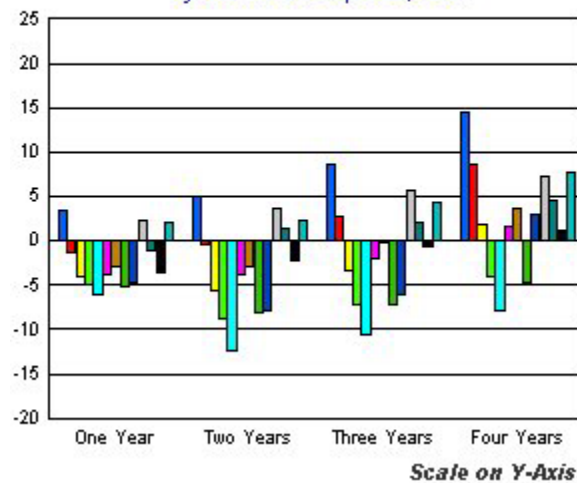
EXAMPLE:

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
```

Enhanced Charting Functionality for Crystal Reports

by Three D Graphics, Inc.



PERSISTENT:

YES

ALSO SEE:

@SCX

@SCX (X-Axis Scale)

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

SYNTAX:

```
@SCX f1 f2
```

PARAMETERS:

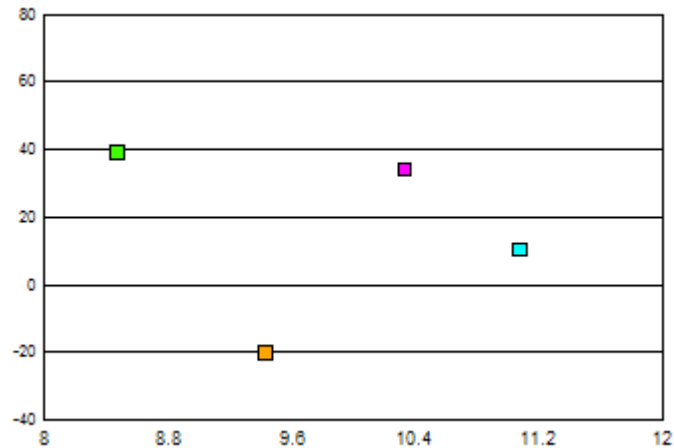
f1; Minimum value to show on the X-Axis

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 (Y2-Axis Scale)

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

SYNTAX:

```
@SCY2 f1 f2
```

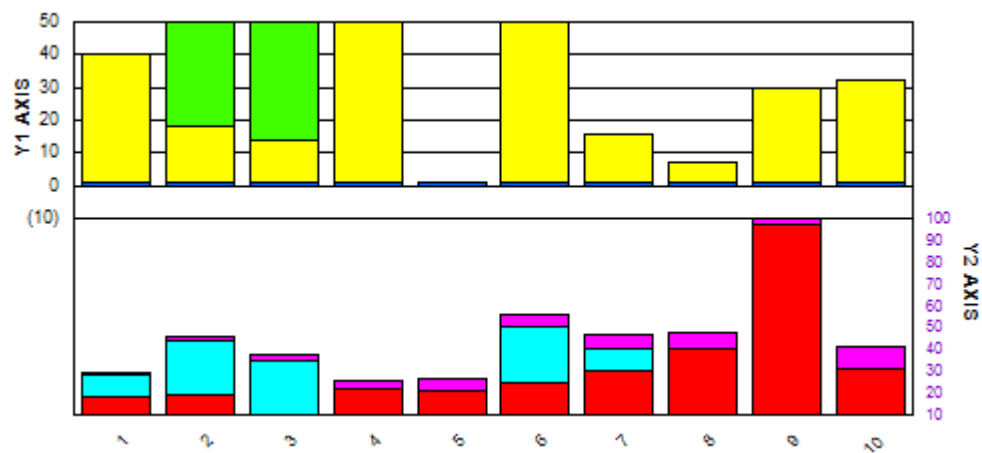
PARAMETERS:

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

@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.

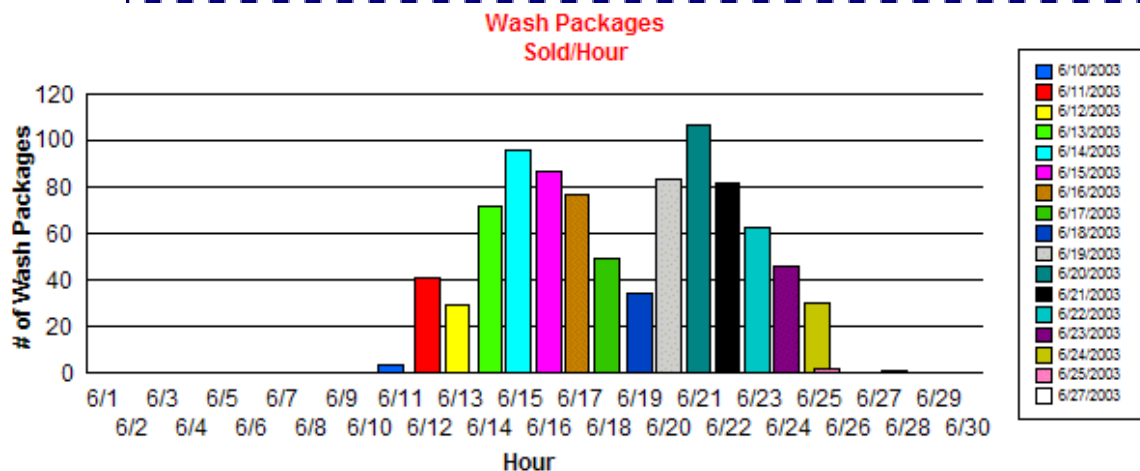
fStop; End point in time. It must be specified in 1901 date format as defined in Microsoft EXCEL.

fInterval; 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 or higher replacement libraries. It will not work in the CR8.x replacement library!

@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:

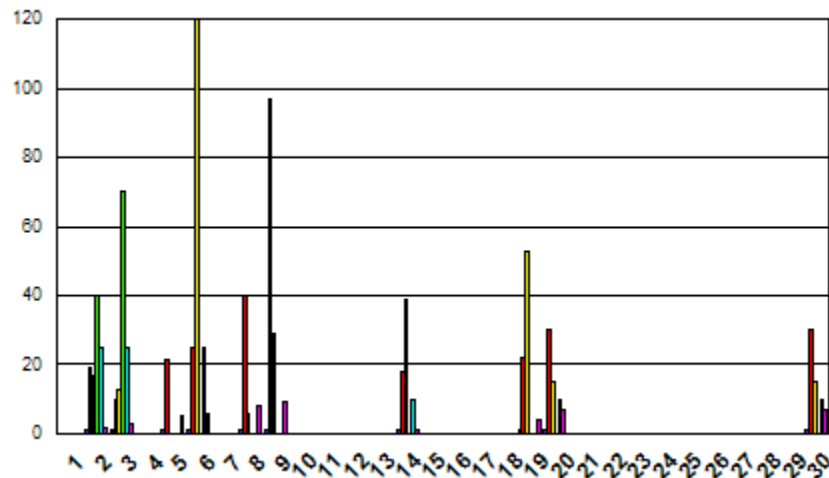
nMode; 0

nStart; Starting group label string

nStop; Ending group label string.

EXAMPLE:

```
@X_AXIS_MODE 0 1 30
```



PERSISTENT:

NO

NOTES:

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

@Y1_FORCE_PERCENT (Y1-Axis Percent Format)

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

SYNTAX:

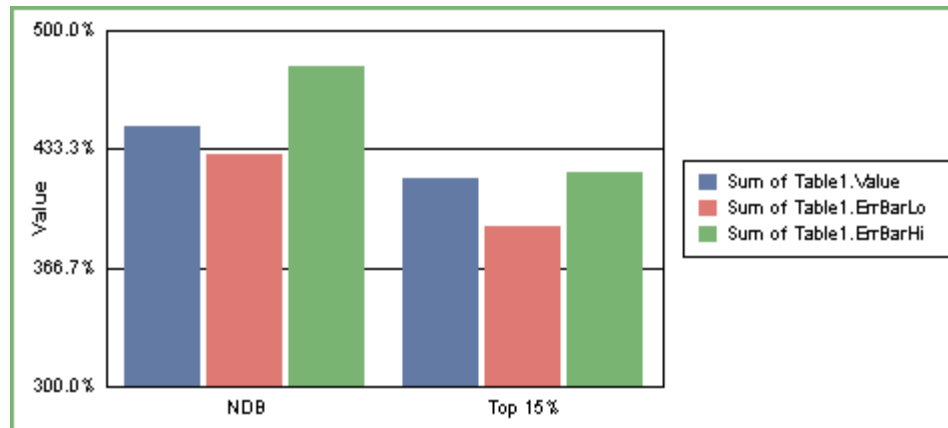
```
@Y1_FORCE_PERCENT bForce
```

PARAMETERS:

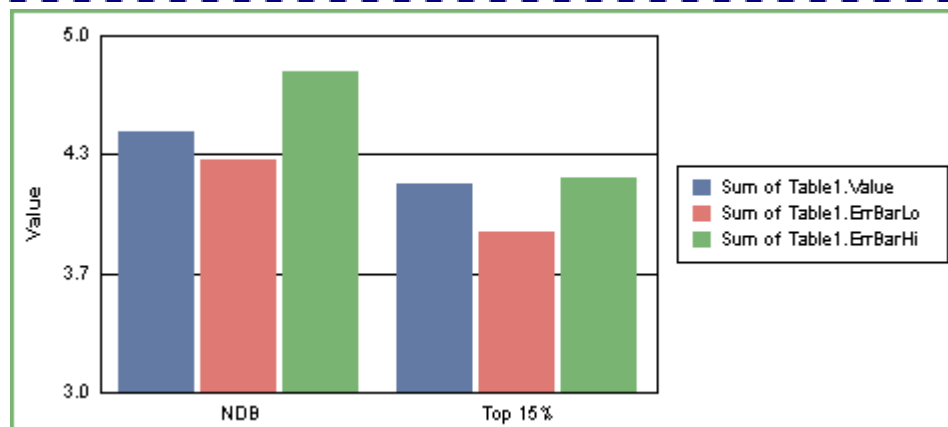
bForce; 0=Draw Y1-axis labels normally. 1=Use percent format.

EXAMPLE:

```
@Y1_FORCE_PERCENT 1
```



```
@Y1_FORCE_PERCENT 0
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@Y1_INVERT (Y1-Axis Invert)

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

SYNTAX:

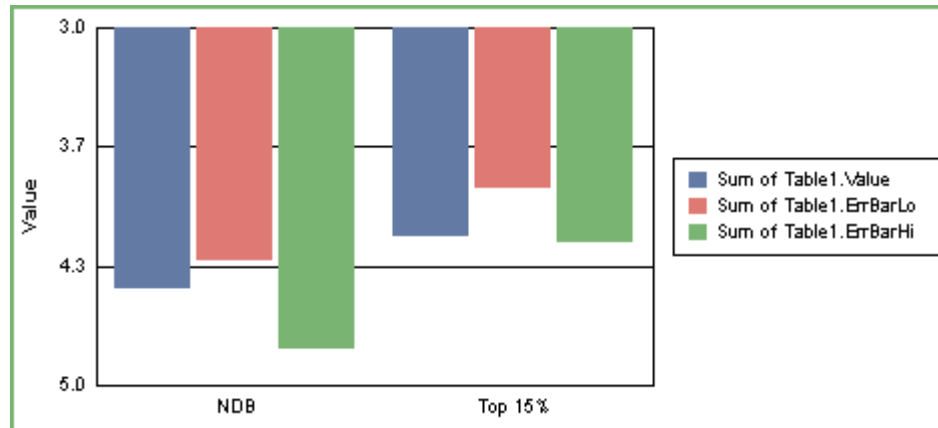
```
@Y1_INVERT bInvert
```

PARAMETERS:

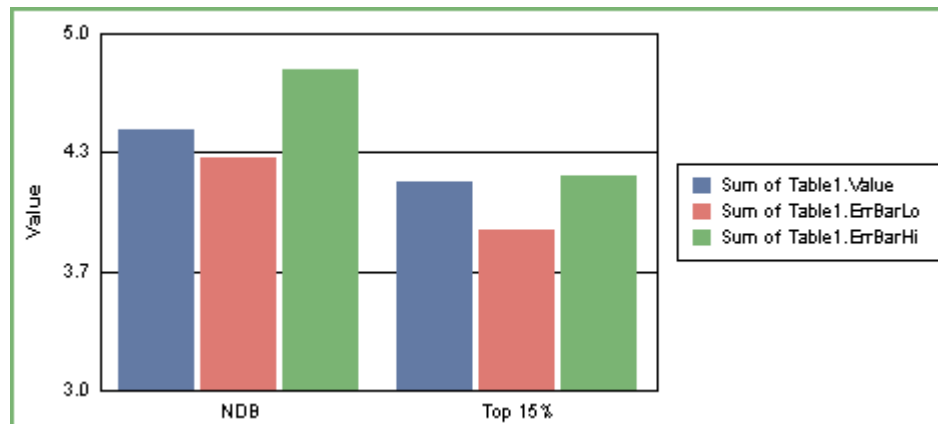
bInvert; 0=Draw values and labels normally. 1=Invert values and labels.

EXAMPLE:

```
@Y1_INVERT 1
```



```
@Y1_INVERT 0
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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
```

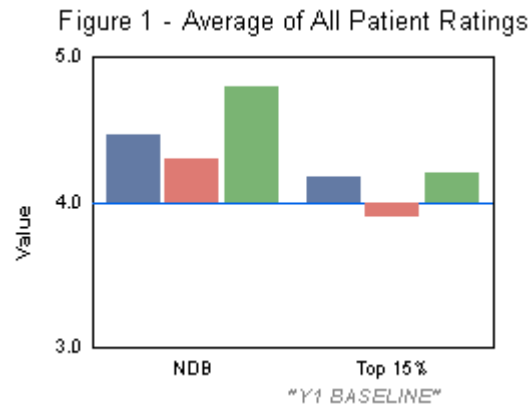
PARAMETERS:

f1; Base line value

EXAMPLE:

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

```
@Y1BASE 4
```



PERSISTENT:

NO

@Y2_FORCE_PERCENT (Y2-Axis Percent Format)

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

SYNTAX:

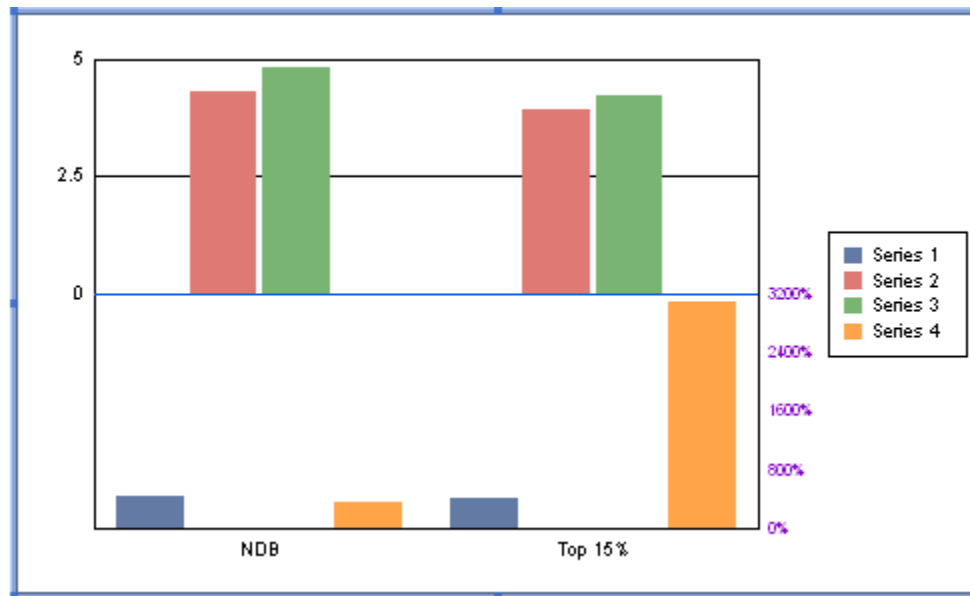
```
@Y2_FORCE_PERCENT bForce
```

PARAMETERS:

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

EXAMPLE:

```
@GRAPHTYPE 18
@Y1_FORCE_PERCENT 0
@Y2_FORCE_PERCENT 1
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 10.x or higher.

@Y2_INVERT (Y2-Axis Invert)

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

SYNTAX:

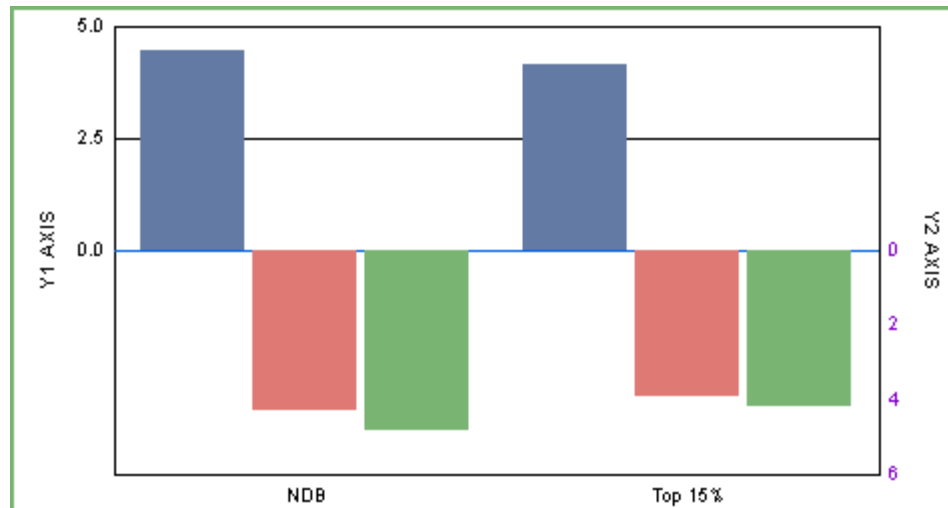
```
@Y2_INVERT bInvert
```

PARAMETERS:

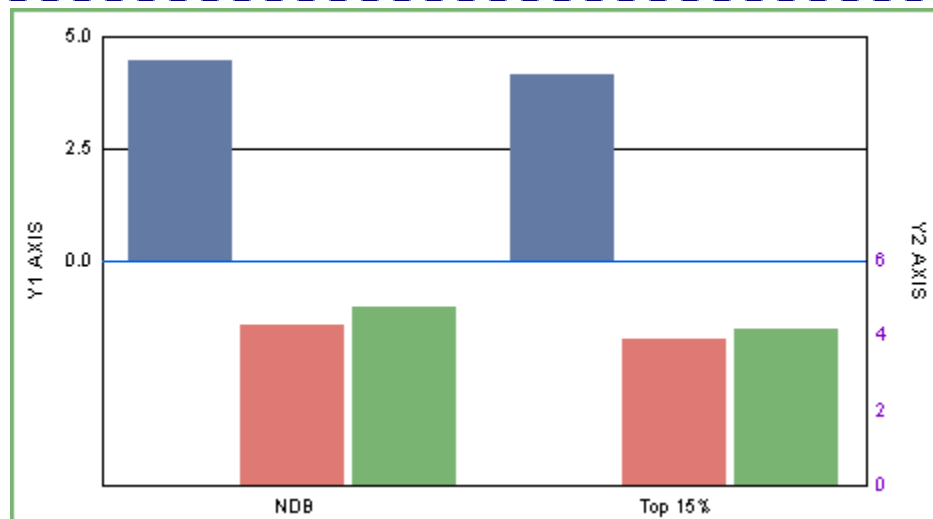
bInvert; 0=Draw values and labels normally. 1=Invert values and labels.

EXAMPLE:

```
@Y2_INVERT 1
```



```
@Y2_INVERT 0
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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:

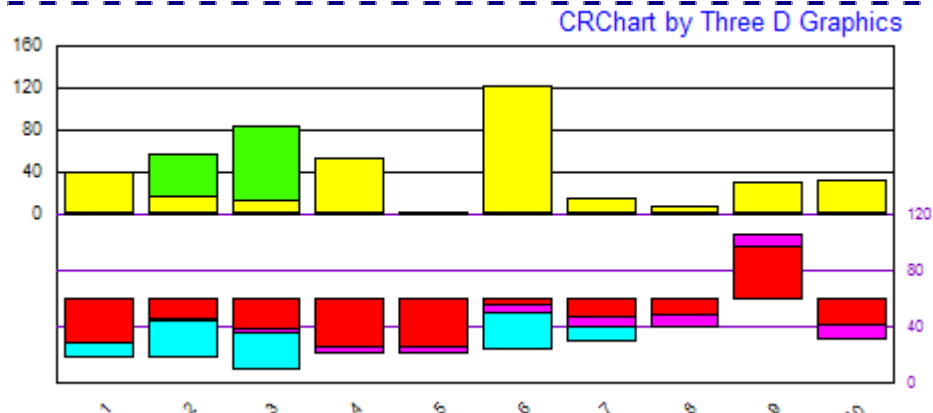
```
@Y2BASE f1
```

PARAMETERS:

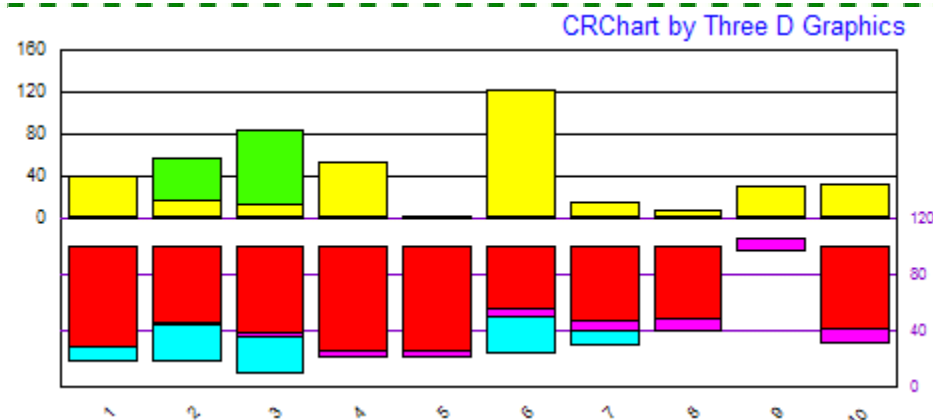
f1; Base line value

EXAMPLE:

```
@Y2BASE 60
```



```
@Y2BASE 100
```



PERSISTENT:

NO

@Y2SLAVE (Y2-Axis Slave to Y1)

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

SYNTAX:

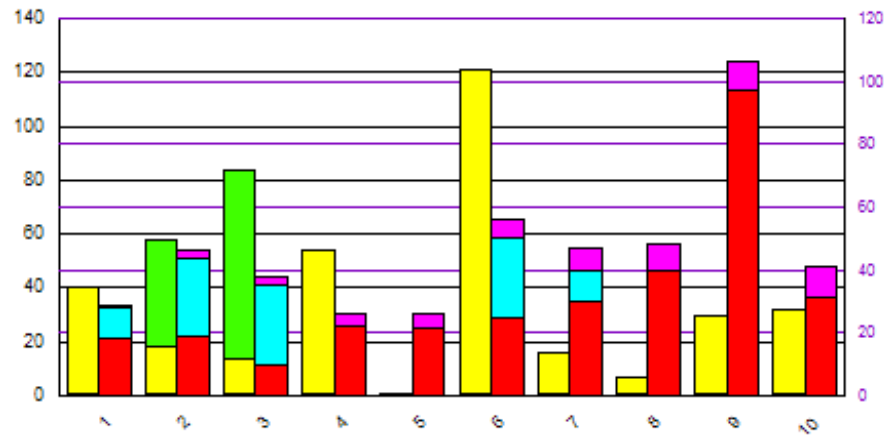
@Y2SLAVE

PARAMETERS:

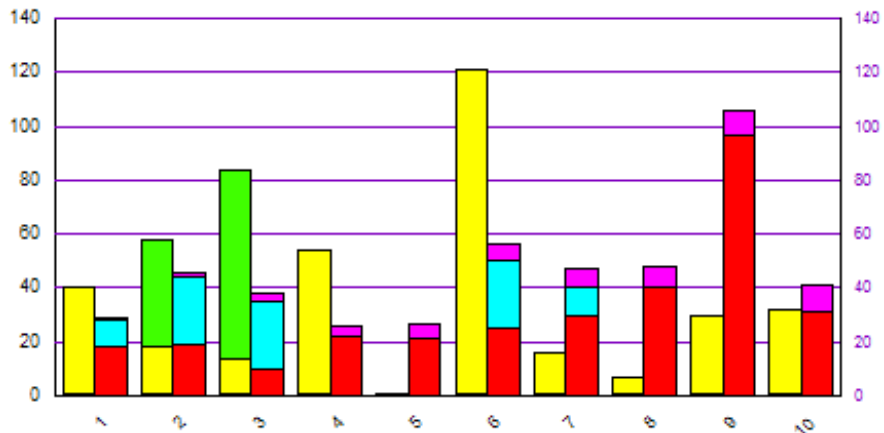
None

EXAMPLE:

Before @Y2SLAVE



After @Y2SLAVE



PERSISTENT:

YES

ALSO SEE:

@Y2SLAVE2

@Y2SLAVE2 (Y1/Y2 Slave to Max Value)

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

SYNTAX:

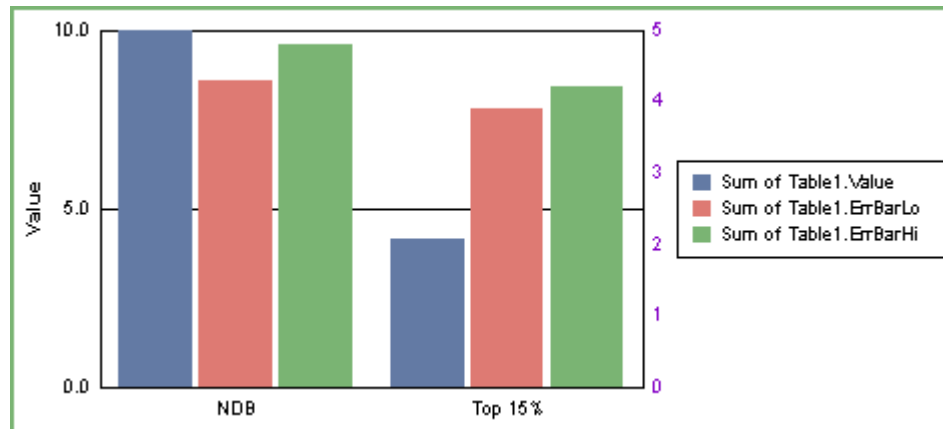
```
@Y2SLAVE2
```

PARAMETERS:

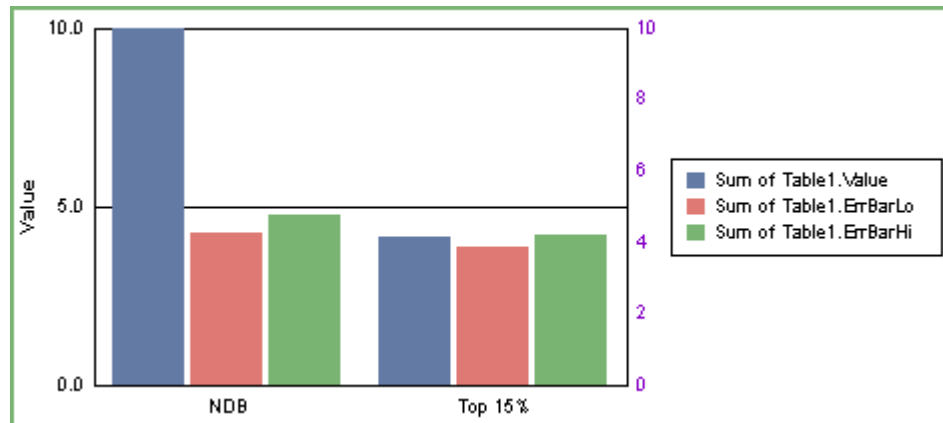
None

EXAMPLE:

Before @Y2SLAVE2



After @Y2SLAVE2



PERSISTENT:

YES

ALSO SEE:

@Y2SLAVE

NOTES:

This macro requires Crystal Reports 9.x or higher.



Section 4: Series & Groups

These macros can be used to control series and groups.

- @FORCE_ABSOLUTE; Force Series to plot Absolute
- @FORCE_SERIES_COUNT; Force the number of Series in a chart
- @FORECAST; Add Blank Groups to a chart
- @GM; Read data in Column Major or Row Major order
- @IG; Ignore a Group
- @IS; Ignore a Series
- @LIMIT_VISIBLE_GROUPS; Limit the number of Visible Groups in a chart
- @PERCENT_SERIES; Create a Ratio Series
- @RG; Reverse Groups
- @RS; Reverse Series
- @SINGLE_GROUP; Draw only the first Group in a Scatter Chart
- @SORT; Sort Series/Groups
- @STEP_LINE; Draw a Series Stepped Line
- @STEP_LINE2; Draw a Series Stepped Line at Values
- @STOP; Force Assign Elements to Series One
- @SWAP; Swap Series/Groups
- @TOTAL_GROUP; Create a Total Group
- @USER_SERIES; Create a User-Defined Series

@FORCE_ABSOLUTE (Force Series Absolute)

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

SYNTAX:

```
@FORCE_ABSOLUTE nSeries bAbs
```

PARAMETERS:

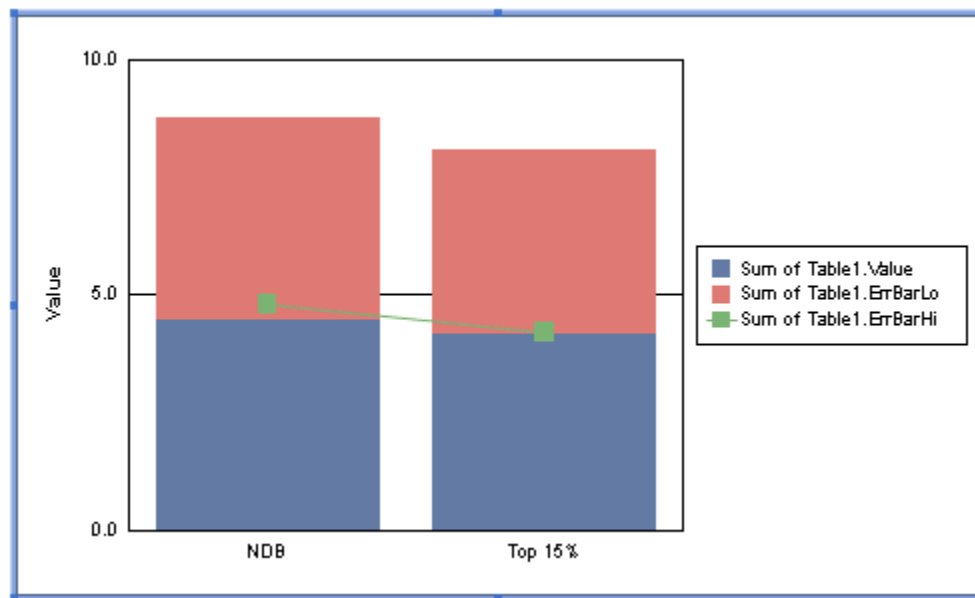
nSeries; 0...999 defines the series to draw absolute (0=Series 1)

bAbs; 1 = Force absolute, 0= Keep Stacked

EXAMPLE:

This example forces the third series to draw ABSOLUTE.

```
@FORCE_ABSOLUTE 2 1
@COMBO 2 1
```



PERSISTENT

NO

NOTES:

This macro requires Crystal Reports 10.x or higher.

@FORCE_SERIES_COUNT (Force Series Count)

This macro forces the number of series in the chart and number of series drawn in the legend to *nElements*. This macro is useful in some cases where the CR viewer sometimes shows more data than the user is expecting.

SYNTAX:

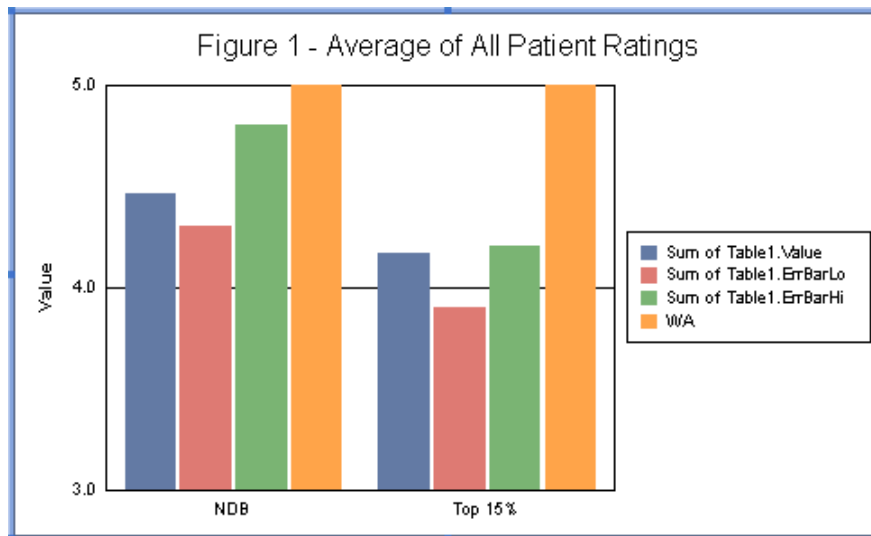
```
@FORCE_SERIES_COUNT nElements
```

PARAMETERS:

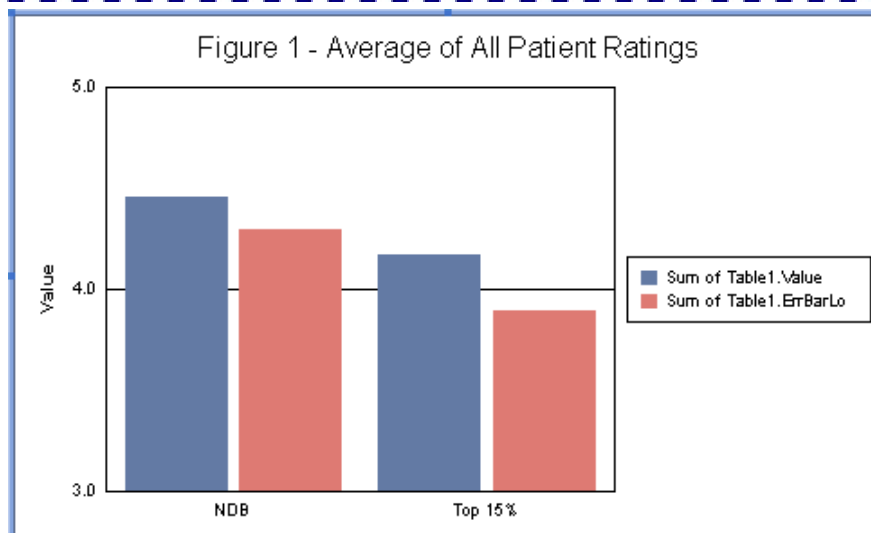
nElements; 0...1024 defines the number of series to draw

EXAMPLE:

```
@FORCE_SERIES_COUNT 4
```



```
@FORCE_SERIES_COUNT 2
```



PERSISTENT

NO

NOTES:

This macro requires Crystal Reports 10.x or higher.

@FORECAST (Add Blank Groups)

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

SYNTAX:

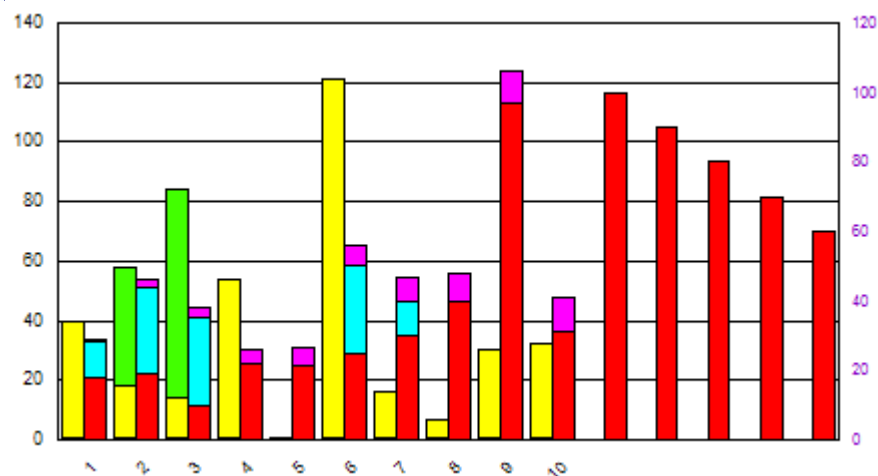
```
@FORECAST nPeriods
```

PARAMETERS:

nPeriods; 0...1024 groups

EXAMPLE:

```
@FORECAST 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)

@GM (Group Major)

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

SYNTAX:

```
@GM n
```

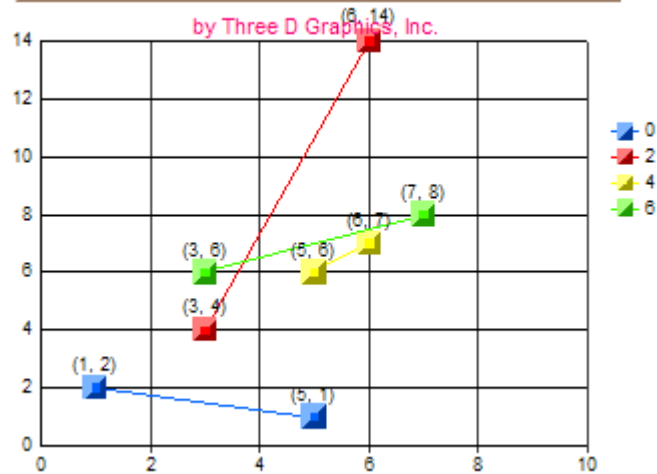
PARAMETERS:

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

EXAMPLE:

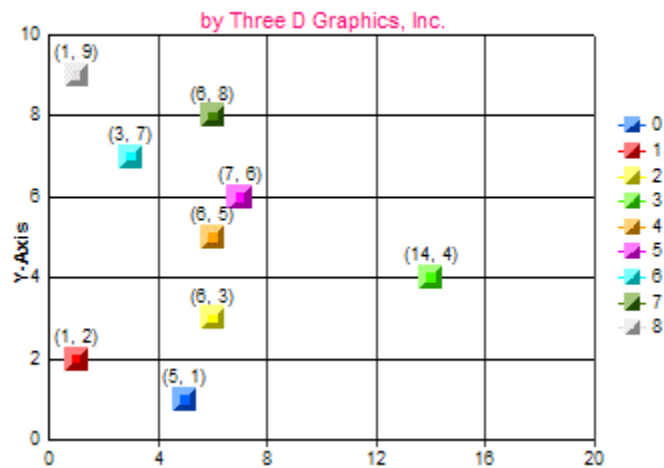
```
@GM 0
```

Enhanced Charting Functionality for Crystal Reports



```
@GM 1
```

Enhanced Charting Functionality for Crystal Reports



PERSISTENT:

NO

@IG (Ignore Group)

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

SYNTAX:

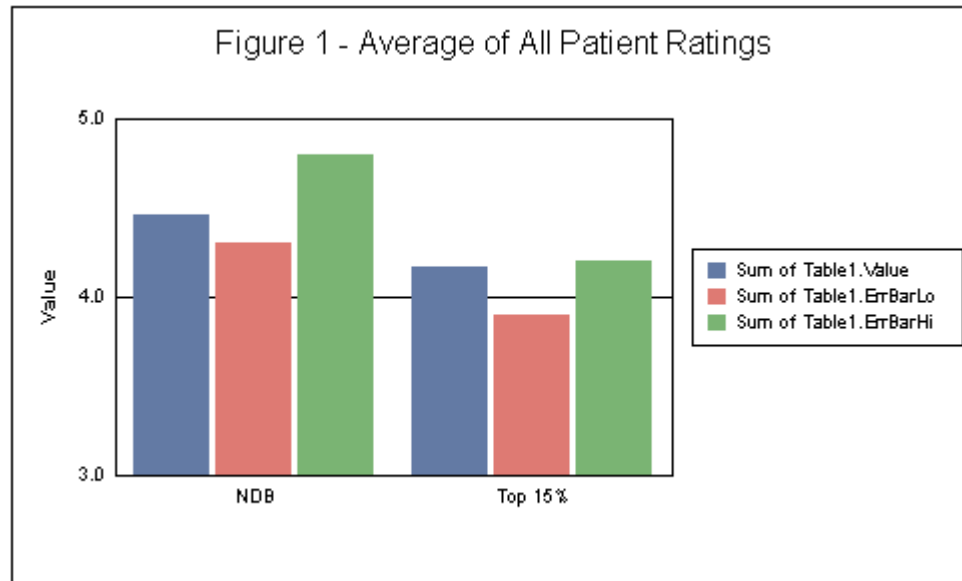
@IG nGroup

PARAMETERS:

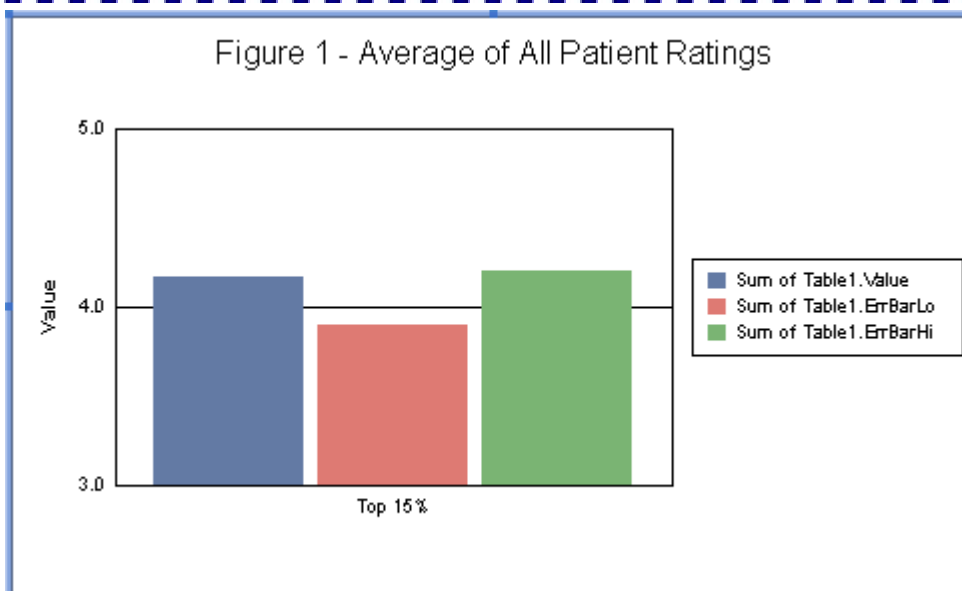
nGroup; Group (0...1024) to hide.

EXAMPLE:

Original chart before @IG 0 Macro:



@IG 0



PERSISTENT:

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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:

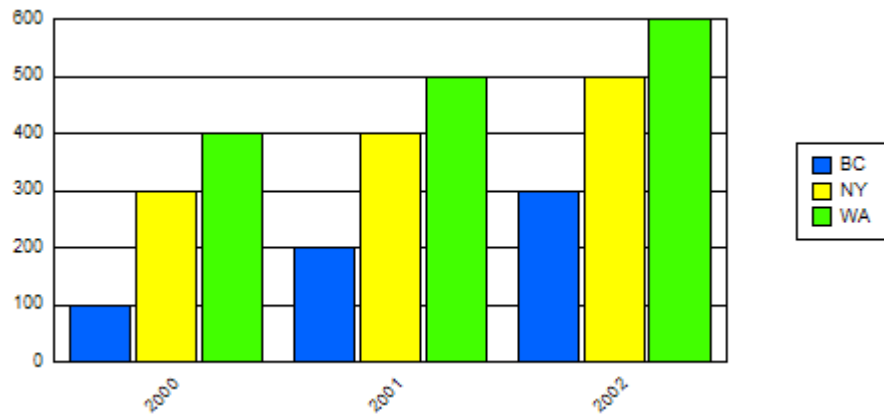
```
@IS n
```

PARAMETERS:

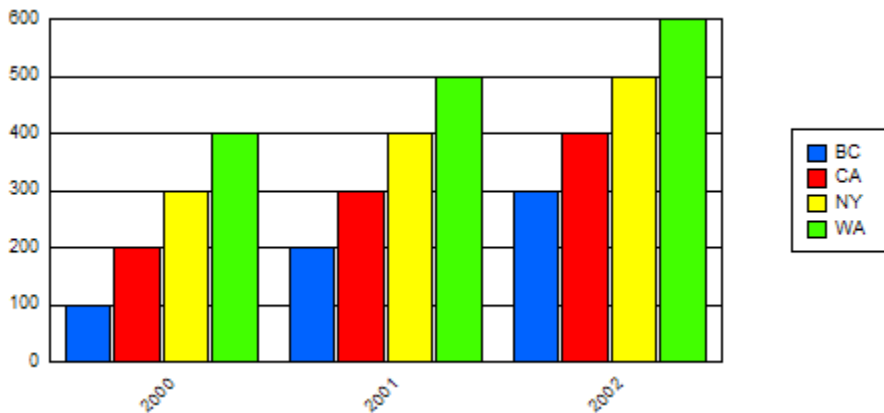
n ; -1... n (where: n = the total number of series in the chart). -1=restore all series that were previously ignored, 0=Series 1, 1=Series 2, etc.

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:

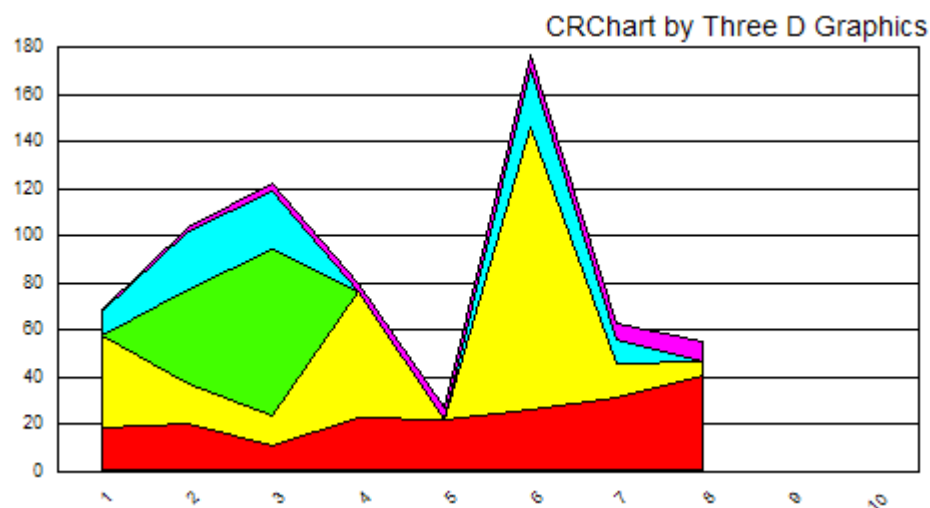
```
@LIMIT_VISIBLE_GROUPS nGroup
```

PARAMETERS:

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

EXAMPLE:

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



PERSISTENT:

YES

NOTES:

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

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

@PERCENT_SERIES (Create Ratio Series)

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

SYNTAX:

```
@PERCENT_SERIES nTop, nBottom, nTarget
```

PARAMETERS:

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

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

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

EXAMPLE:

```
@AGL 0 Los Angeles~ @AGL 1 California
```

```
@PERCENT_SERIES 0 1 2
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@RG (Reverse Group)

This macro reverses the order of groups in a chart.

SYNTAX:

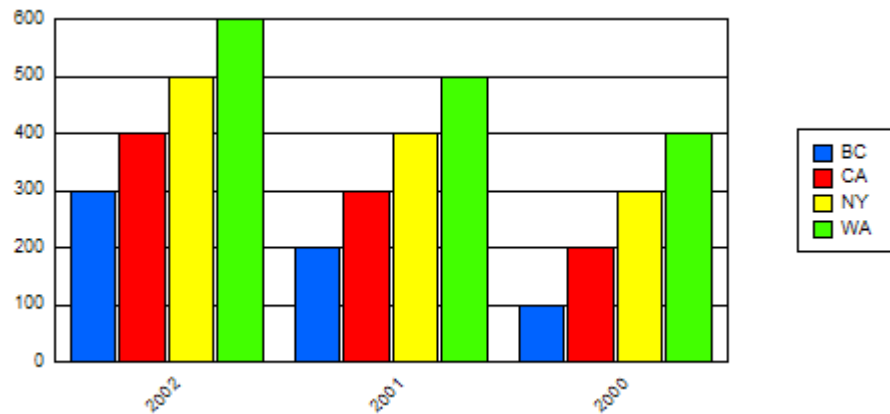
@RG *n*

PARAMETERS:

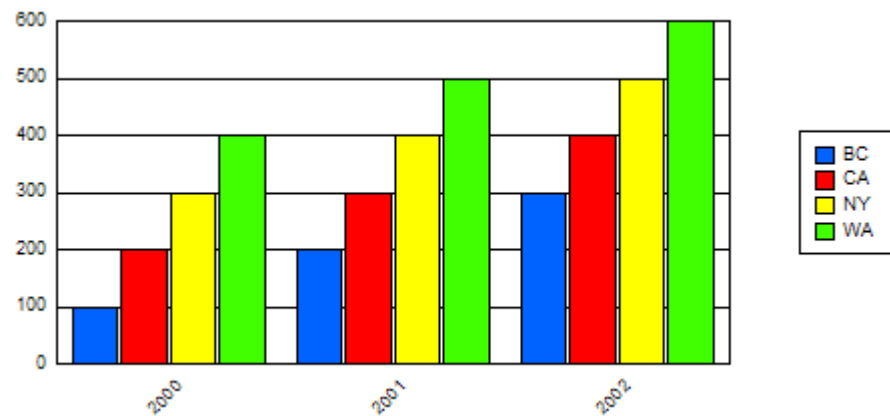
n; 1=true/reverse groups, 0=false/do not reverse groups.

EXAMPLE:

@RG 1



@RG 0



PERSISTENT:

YES

ALSO SEE:

@RS

@RS (Reverse Series)

This macro reverses the order of series in a chart.

SYNTAX:

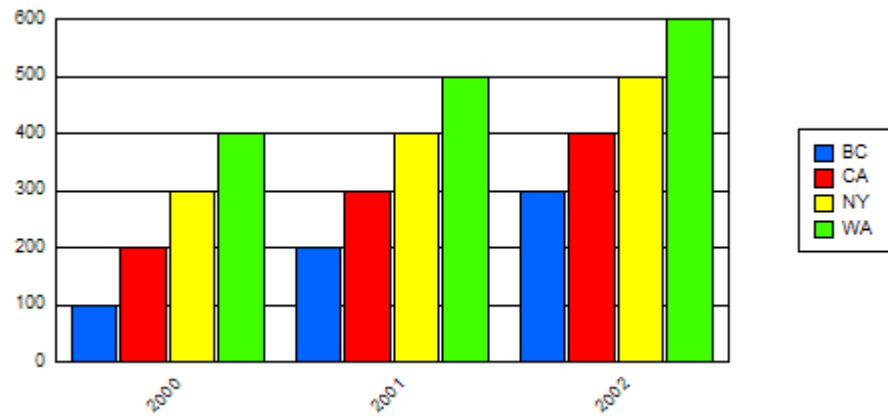
```
@RS n
```

PARAMETERS:

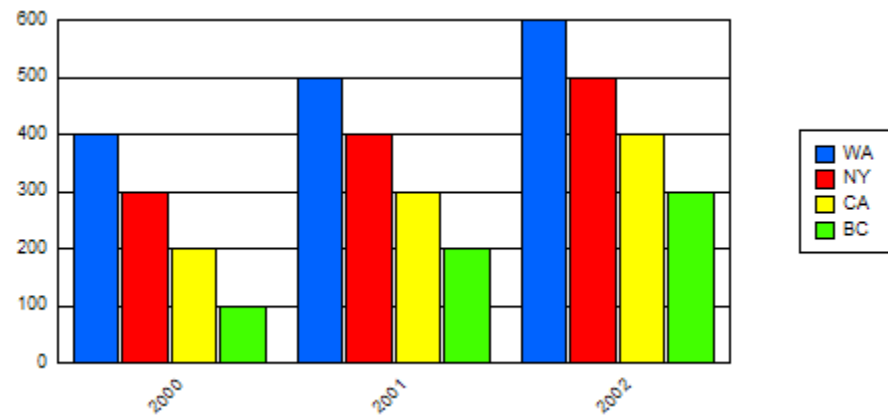
n; 1=true/reverse series, 0=false/do not reverse series.

EXAMPLE:

```
@RS 0
```



```
@RS 1
```



PERSISTENT:

YES

ALSO SEE:

@RG

@SINGLE_GROUP (Single Group in Scatter Chart)

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

SYNTAX:

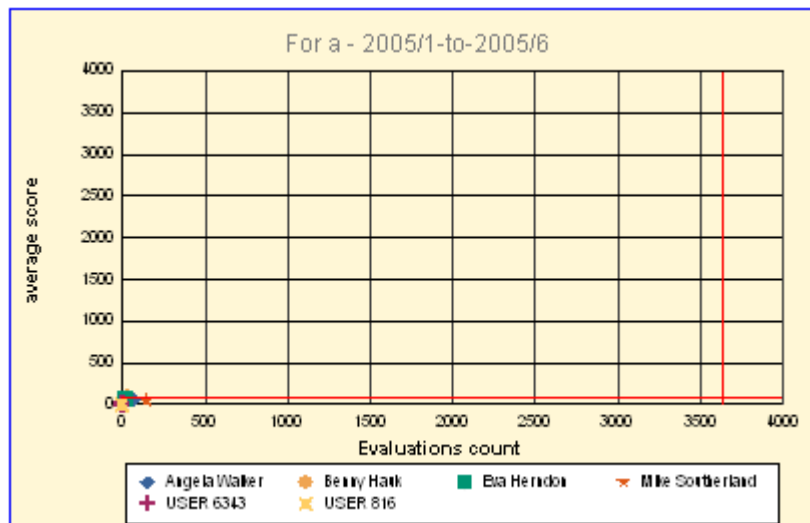
```
@SINGLE_GROUP
```

PARAMETERS:

None

EXAMPLE:

```
@Y P2  
@X P3  
@SINGLE_GROUP
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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:

0 = Sort Series Labels in Alphabetical order (a...z)

1 = Sort Series Labels in Reverse Alphabetical order (z...a)

2 = Sort Groups Labels in Alphabetical order (a...z)

3 = Sort Groups Labels in Reverse Alphabetical order (z...a)

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

5 = Sort Series Totals in descending numeric value

6 = Sort Groups Totals in ascending numeric value

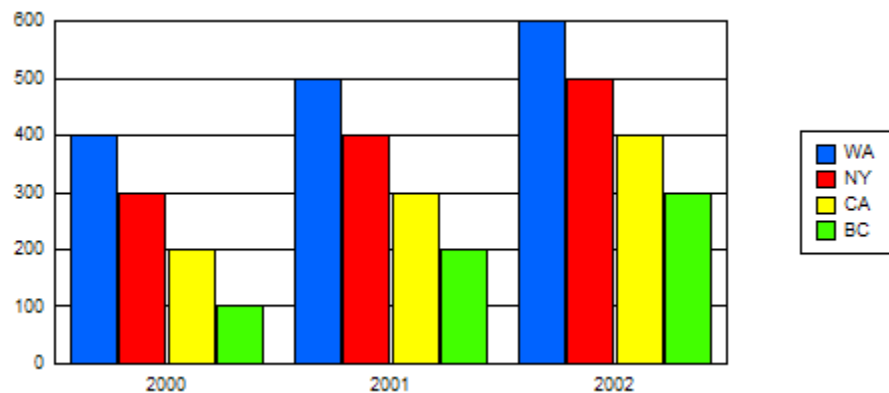
7 = Sort Groups Totals in descending numeric value

8 = First Series Key to Groups Ascending

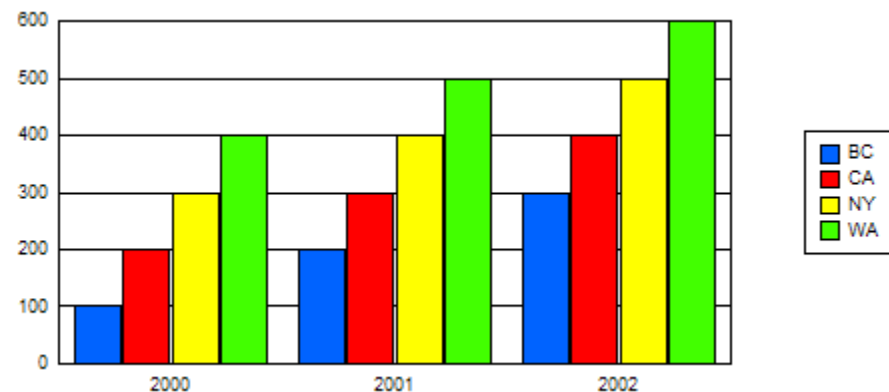
9 = First Series Key to Groups Descending

EXAMPLE:

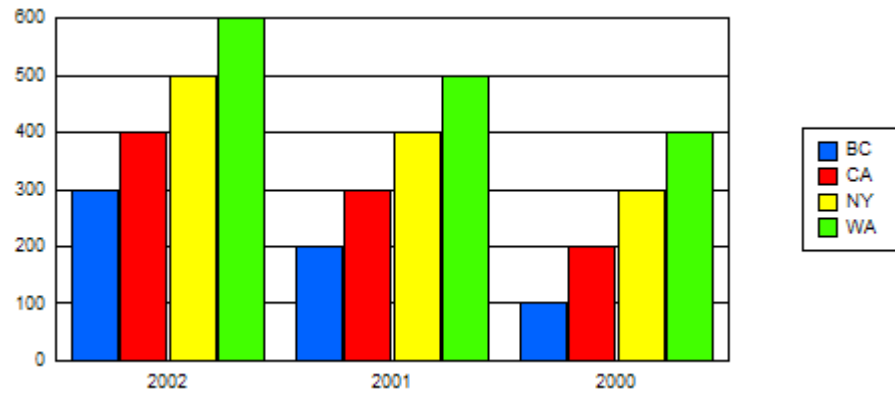
```
@SORT 1
```



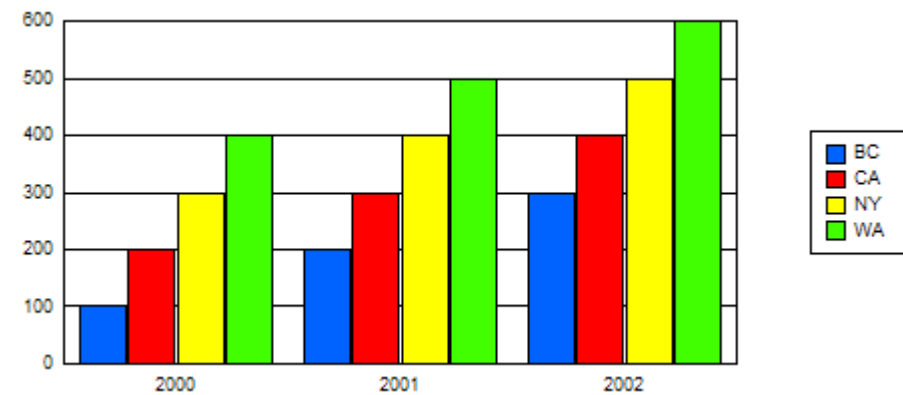
```
@SORT 2
```



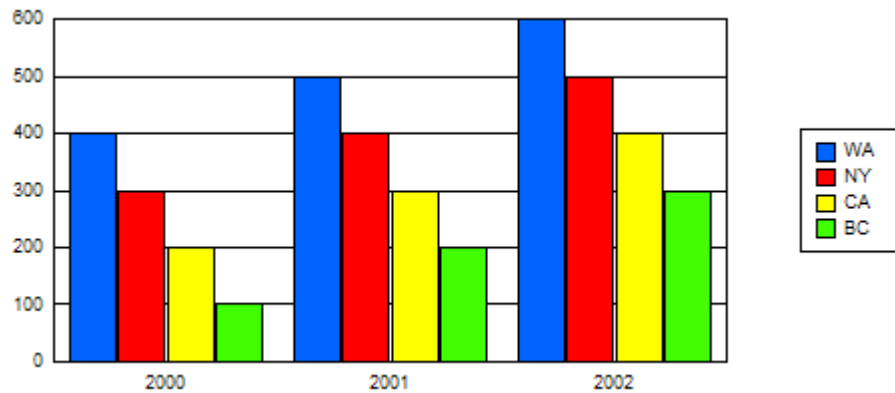
@SORT 3



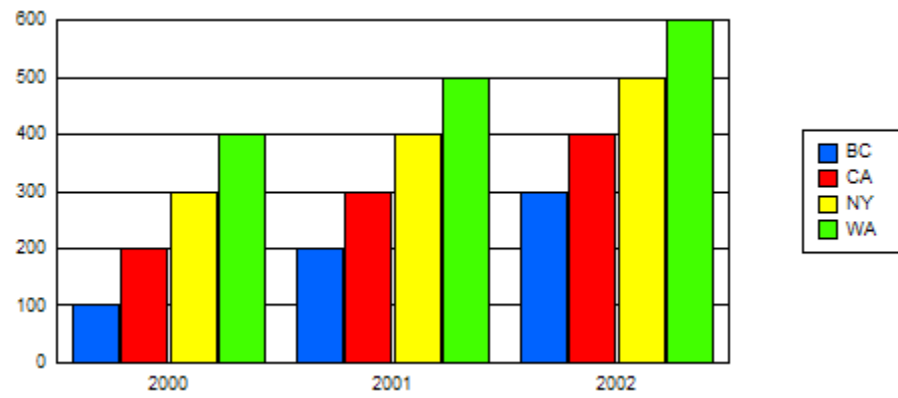
@SORT 4



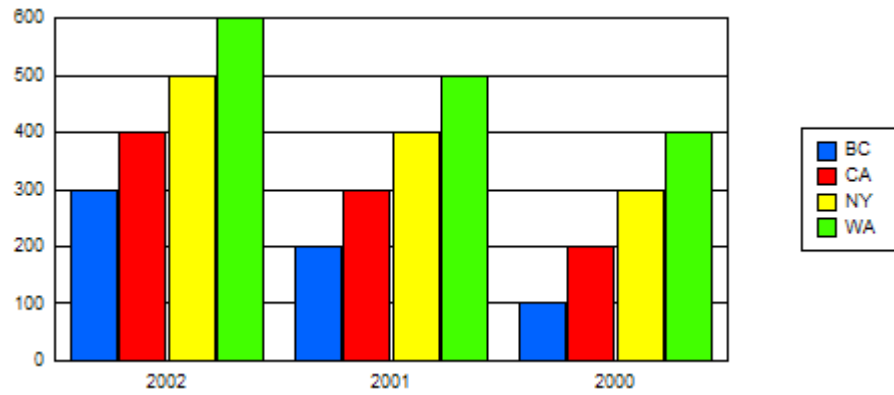
@SORT 5



@SORT 6



@SORT 7



PERSISTENT:
NO

@STEP_LINE (Series Stepped Line)

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

SYNTAX:

```
@STEP_LINE nSeries bStep
```

PARAMETERS:

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

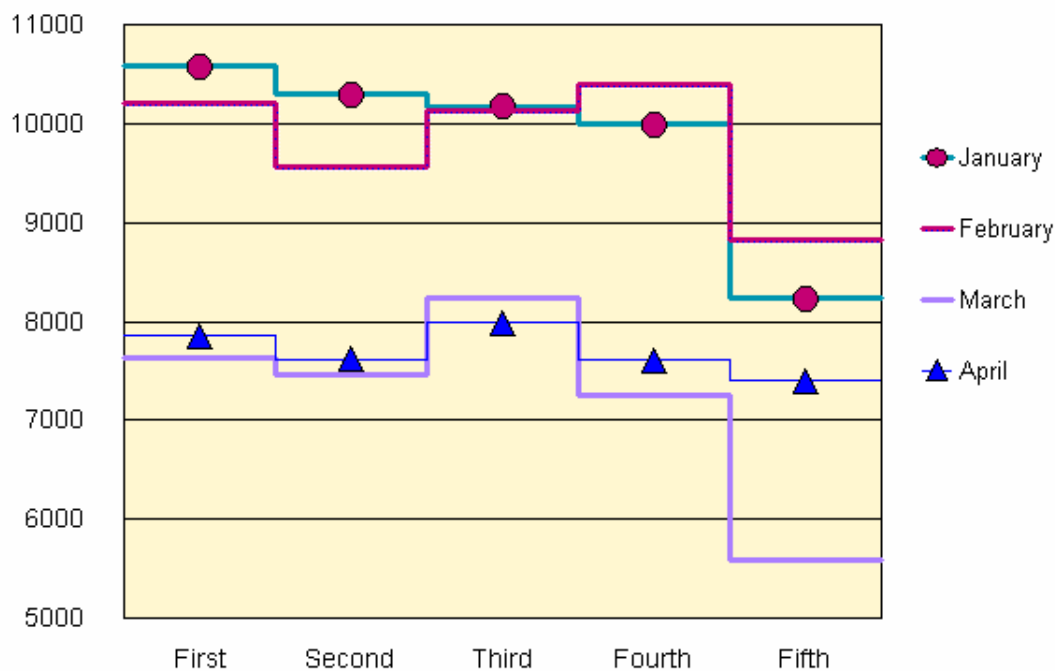
bStep; 1 = draw *nSeries* as stepped line. 0 = draw *nSeries* normally.

EXAMPLE:

```
@STEP_LINE 1 1
```

```
@STEP_LINE 2 1
```

CRChart does StepLines!



PERSISTENT:

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.

@STEP_LINE2 (Series Stepped Line at Values)

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

SYNTAX:

```
@STEP_LINE2 nSeries bStep
```

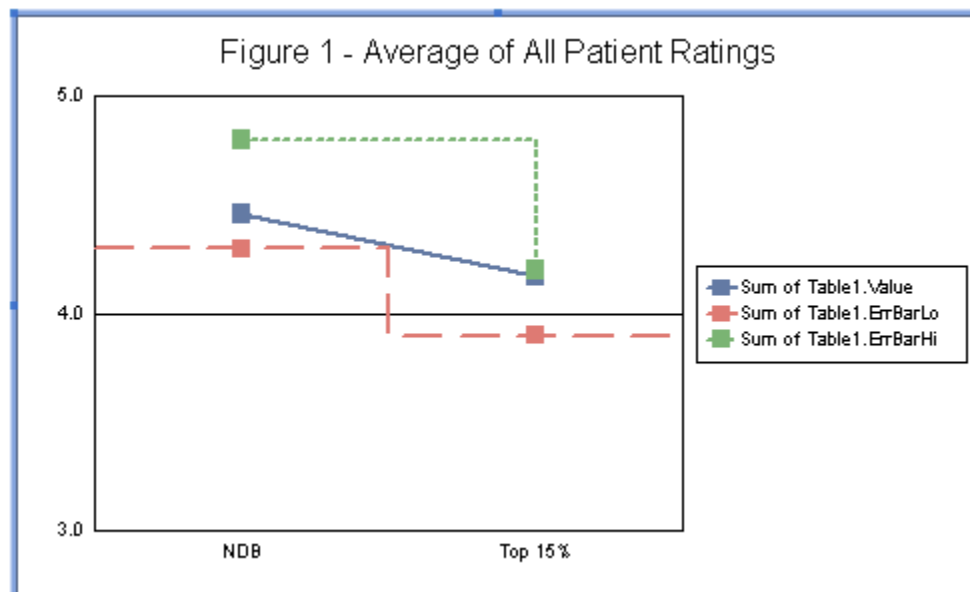
PARAMETERS:

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

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

EXAMPLE:

```
@GRAPHTYPE 28
@STEP_LINE 1 1
@STEP_LINE2 2 1
@LS 0 250 0
@LS 1 250 1
@LS 2 250 2
```



PERSISTENT:

YES

NOTES:

This macro requires Crystal Reports 11.x or higher.

@STOP (Force Assign Elements to Series One)

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

SYNTAX:

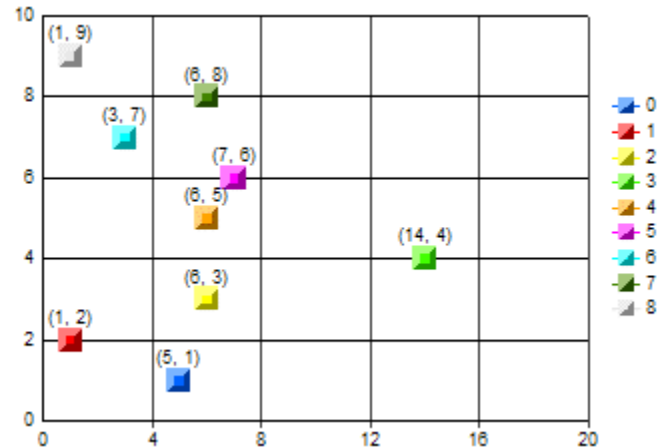
```
@STOP n
```

PARAMETERS:

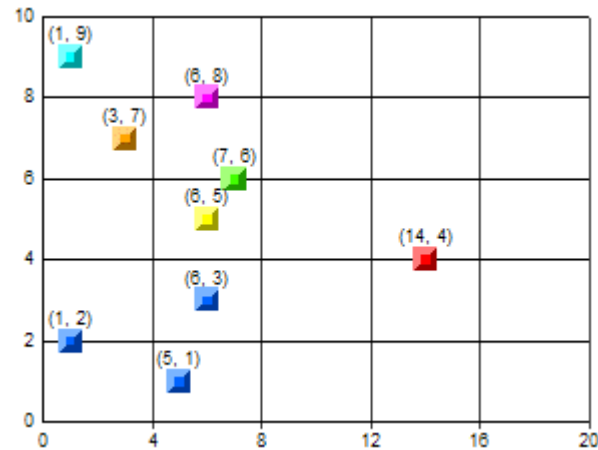
n; number of elements to force assign to first series.

EXAMPLE:

Before @STOP applied



```
@STOP 3
```



PERSISTENT:

NO

@SWAP (Swap Series/Groups)

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

SYNTAX:

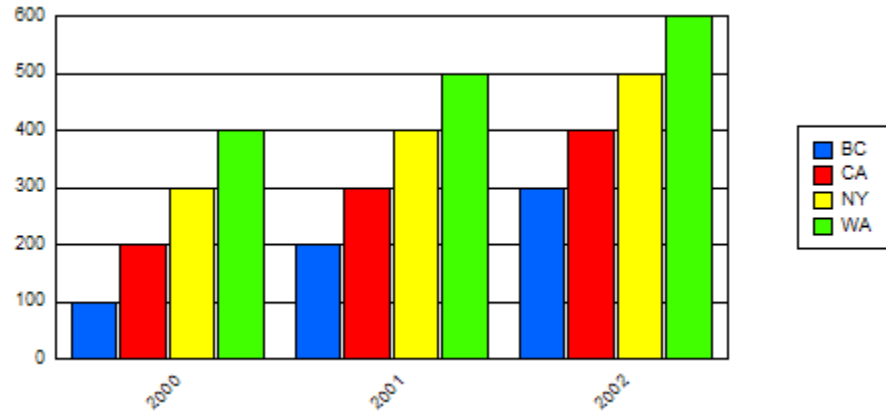
```
@SWAP n
```

PARAMETERS:

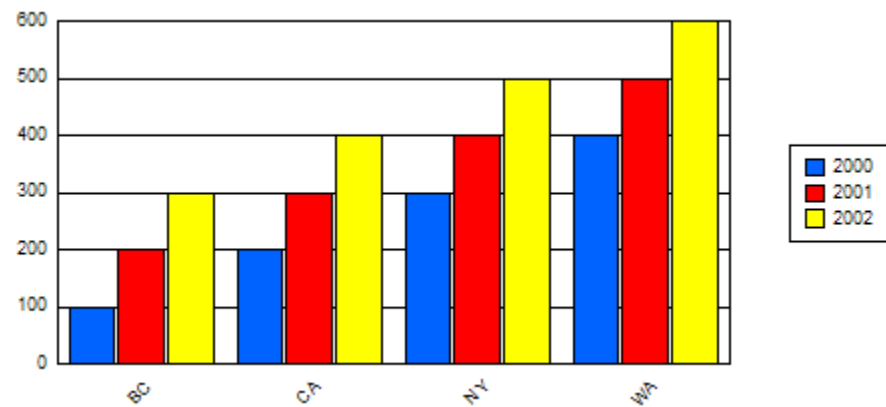
n; 1 = swap series/group orientation, 0= do not swap series/group orientation

EXAMPLE:

```
@SWAP 0
```



```
@SWAP 1
```



PERSISTENT:

YES

@TOTAL_GROUP (Create a Total Group)

This macro adds an extra group to a chart and populates it with risers that are equal to the total of all groups that came before it. The optional label parameter (sz) can be used to label the group on the X-axis.

SYNTAX:

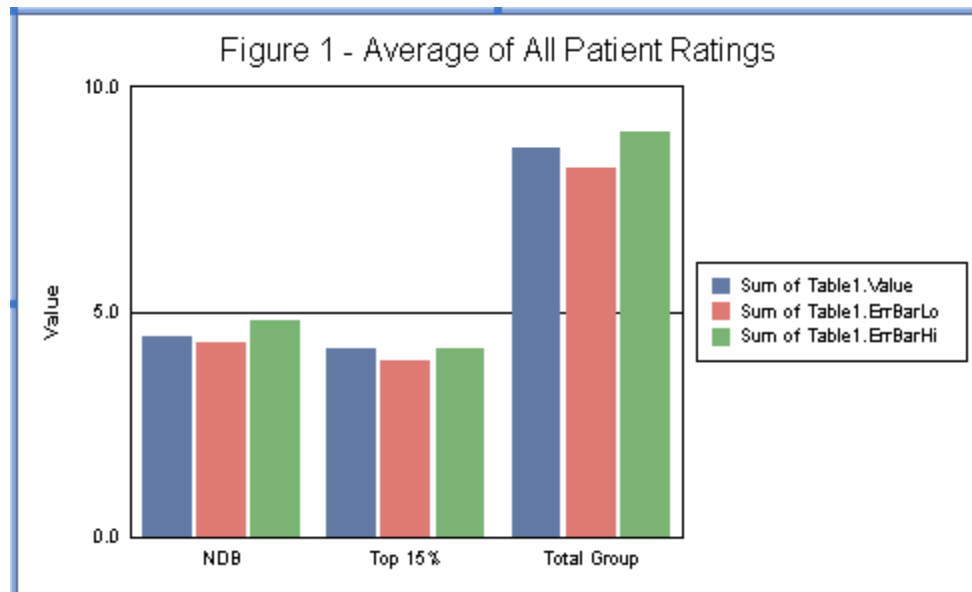
```
@TOTAL_GROUP sz
```

PARAMETERS:

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

EXAMPLE:

```
@TOTAL_GROUP Total Group
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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.

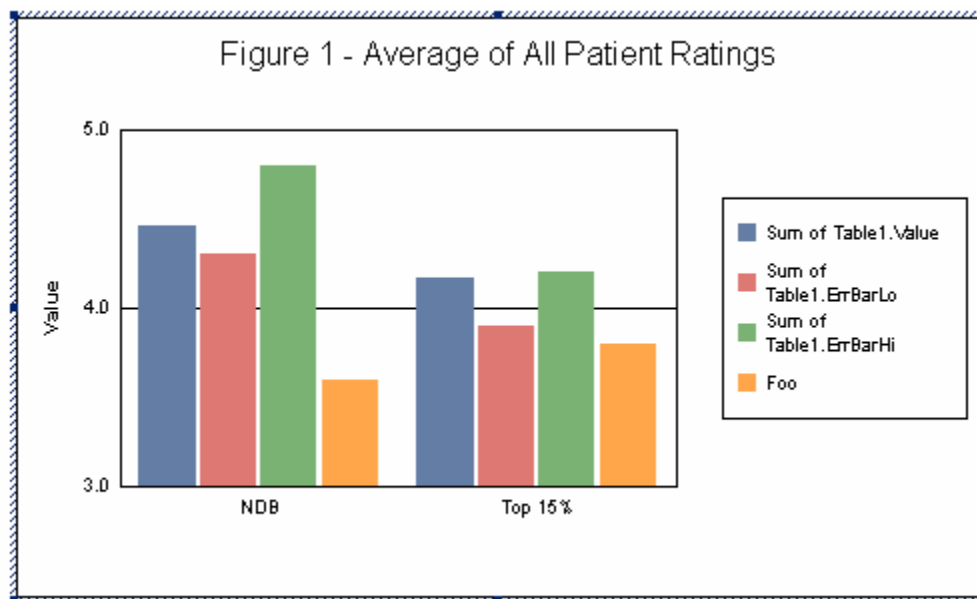
fValue1 fValue2 ... fValueN; Values to assigned to each *nElements*.

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

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



Section 5: Labels

Use these macros to manage series and group labels.

- @AGL; Alias/Change a Group Label
- @ASL; Alias/Change a Series Label
- @ASL_DP; Map a sub-string from a label to the legend
- @COND_GROUP_LABEL; Color a riser based on a Group Label
- @COND_GROUP_LABEL2; Color a riser based on a Group Label prefix
- @GROUP_LABELS_ON_BASELINE; Draw Group Labels on Baseline
- @XSKIP; Skip labels on the X-Axis
- @XSKIP2; Skip labels on the X-Axis but Force Last Label

@AGL (*Alias Group Label*)

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

SYNTAX:

```
@AGL n sz
```

PARAMETERS:

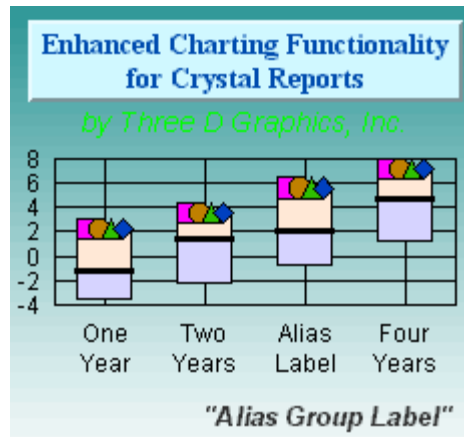
n; 0...number of groups in chart (0=Group 1). Defines the group on which to place the alias label.

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

EXAMPLE:

This example creates a box plot chart with T-Style 'tails' (@BP2), 4 markers (@MK 4) per box plot, 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

@ASL (*Alias Series Label*)

This macro can be used to change a series label.

SYNTAX:

```
@ASL n sz
```

PARAMETERS:

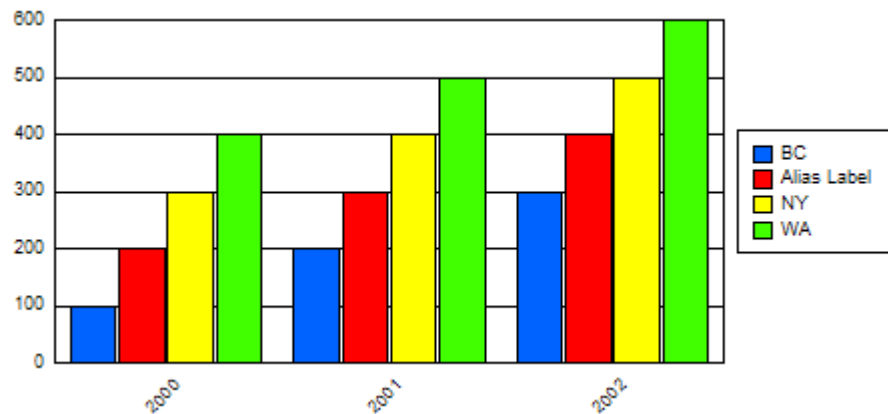
n; 0...number of series in chart (0=Series 1). Defines the series on which to place the alias label.

sz; Optional string. If you intend to define another macro in the same title field, terminate the label string 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 tilde (~), this macro maps the sub-string to the left of the first tilde to the series (legend) label specified by *nS*. When group labels are drawn, the sub-string and tilde are stripped out so that neither appear on the group axis.

SYNTAX:

```
@ASL_DP nS
```

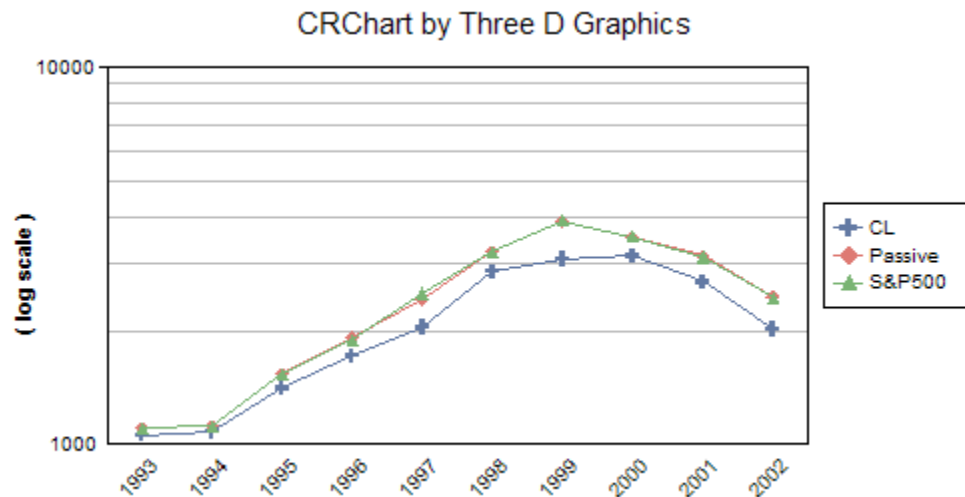
PARAMETERS:

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

@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:

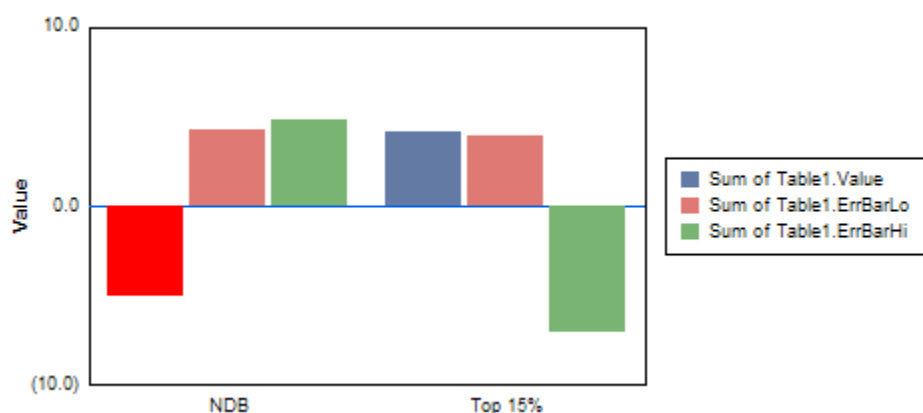
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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

SYNTAX:

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

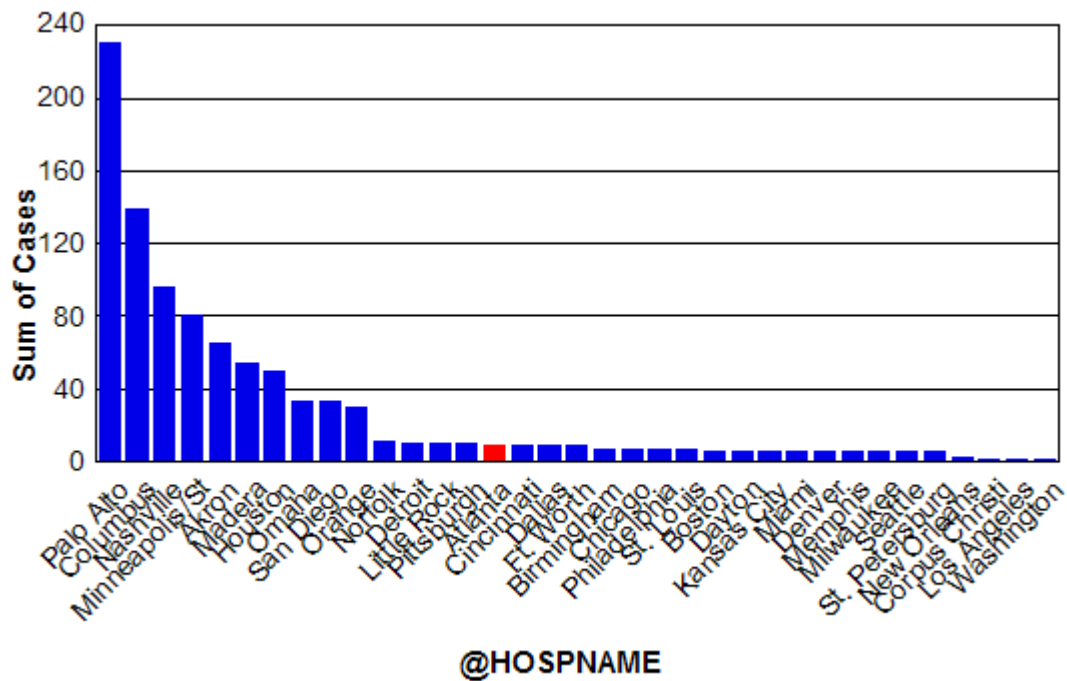
PARAMETERS:

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

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

EXAMPLE:

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



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

PERSISTENT:

NO

@XSKIP (X-Axis Skip Labels)

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

SYNTAX:

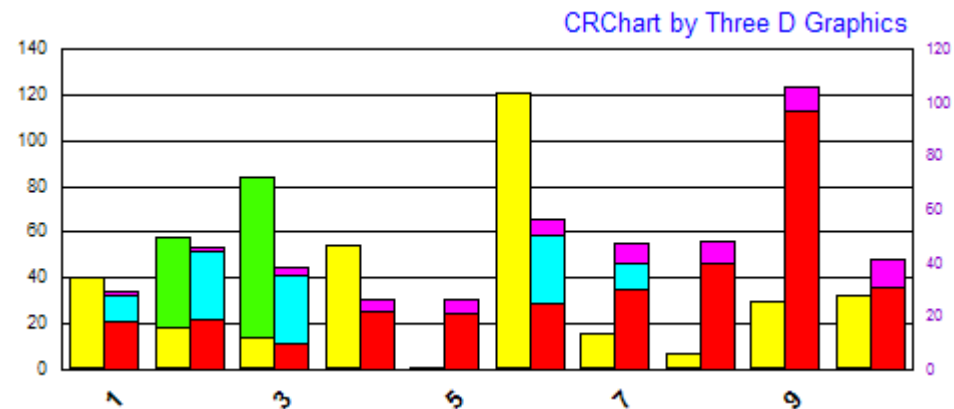
@XSKIP value

PARAMETERS:

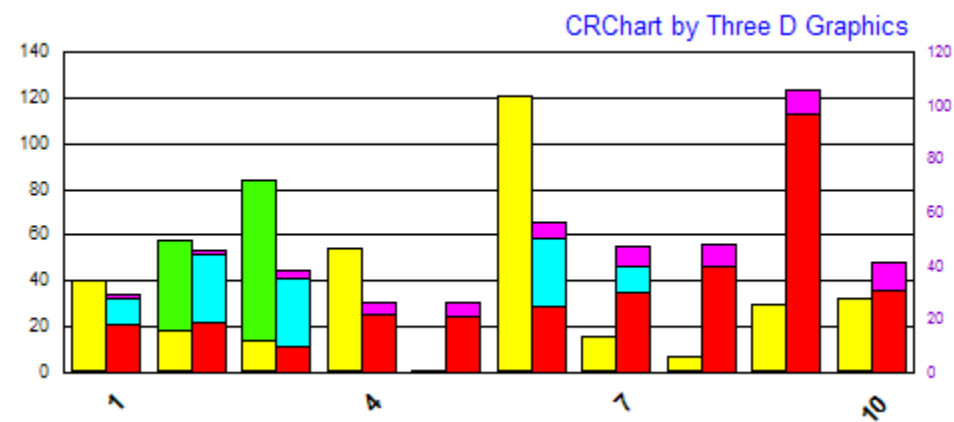
value; skip value

EXAMPLE:

@XSKIP 2



@XSKIP 3



PERSISTENT:

NO

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

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

SYNTAX:

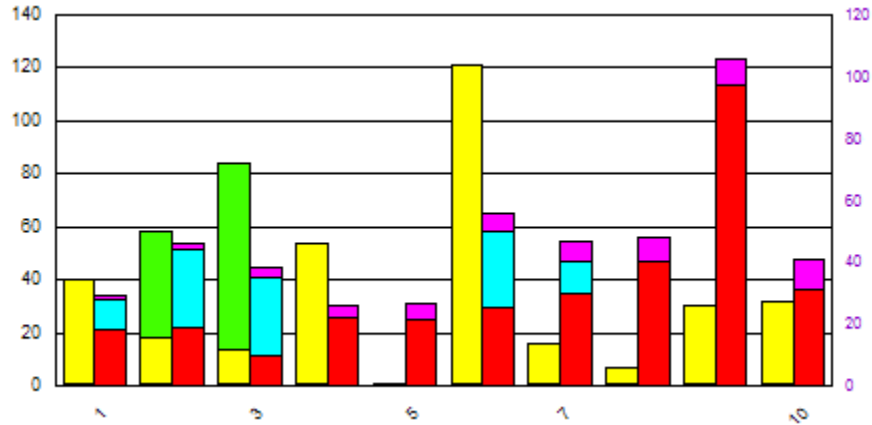
```
@XSKIP2 value
```

PARAMETERS:

value; skip value

EXAMPLE:

```
@XSKIP2 2
```



PERSISTENT:

NO

Section 6: Riser & Markers

Use these macros to control and format risers and markers:

- @HIDE_ZERO; Hide Risers/Markers for zero data points
- @HL; Highlight a Riser/Marker
- @MARKER; Define Marker Shapes
- @MCOLOR; Define Marker Colors
- @PAT; Apply a Pattern to Riser
- @RISER_BORDER; Enable/Disable Riser Borders
- @RISER_OVERLAP; Define the amount of Overlap between risers in side-by-side/clustered bar charts
- @RISER_WIDTH; Define the Width of Risers in bar charts

Also see the @SZ macro in “Macros for Box Plots” (Section 11) to set the size of markers in scatter charts.

@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:

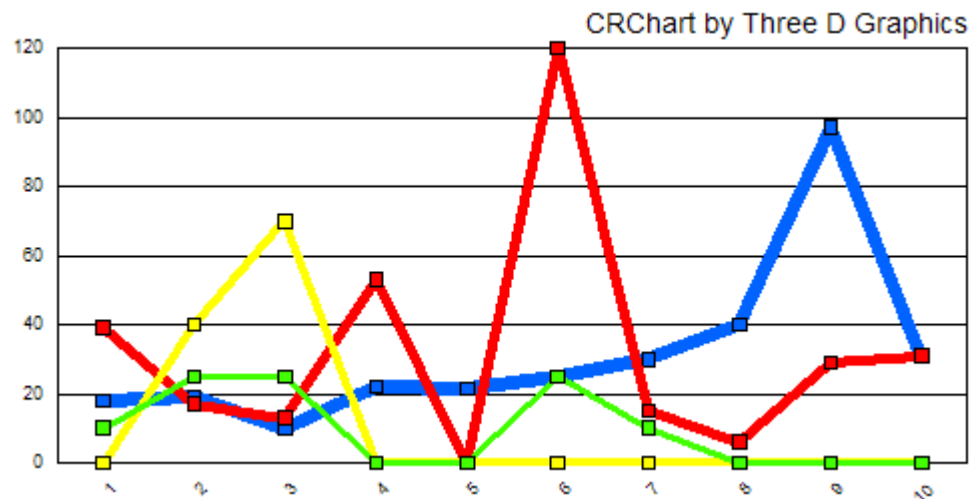
```
@HIDE_ZERO
```

PARAMETERS:

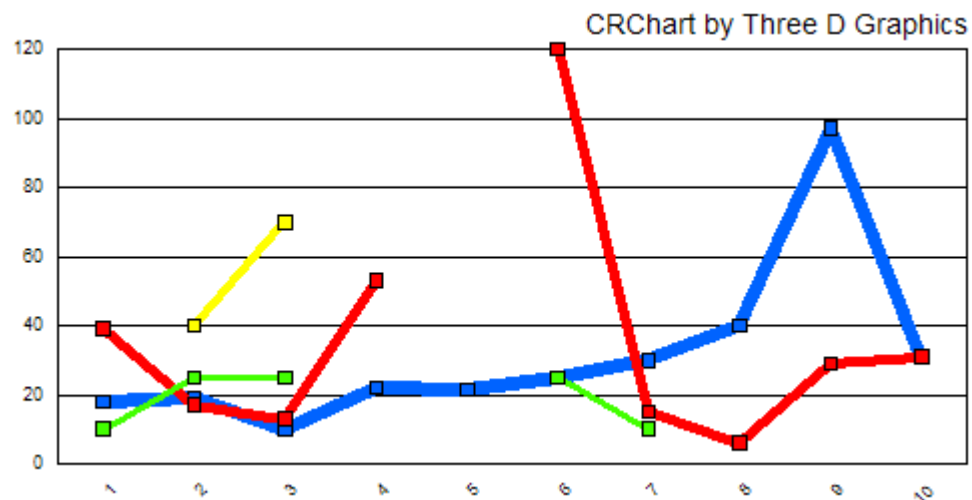
None

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:

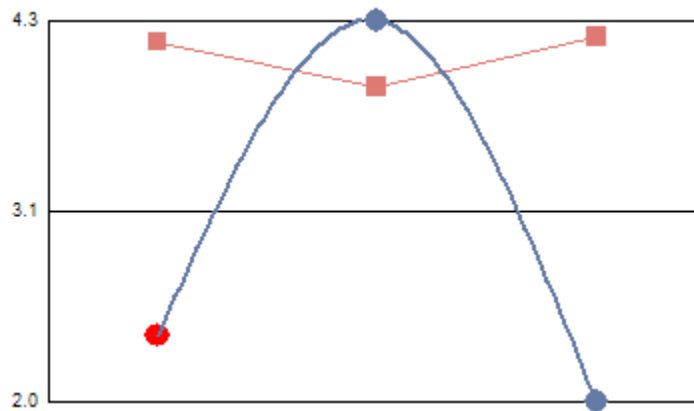
nSeries; Series #

nGroup; Group #

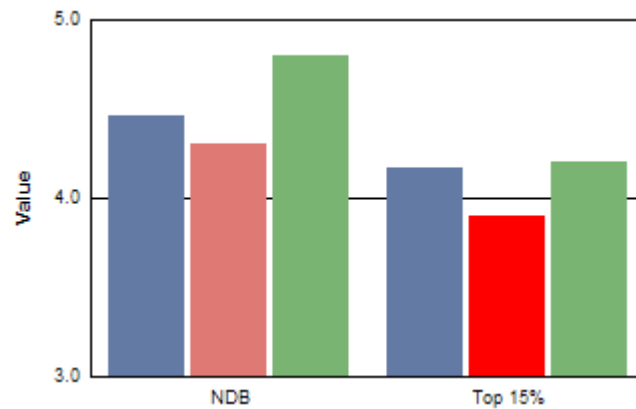
nRed, *nGreen*, *nBlue*: 0...255 specifies the RGB value to be applied to the riser/marker at *nSeries* and *nGroup*.

EXAMPLE:

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



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



PERSISTENT:

YES

@MARKER (Marker Shapes)

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

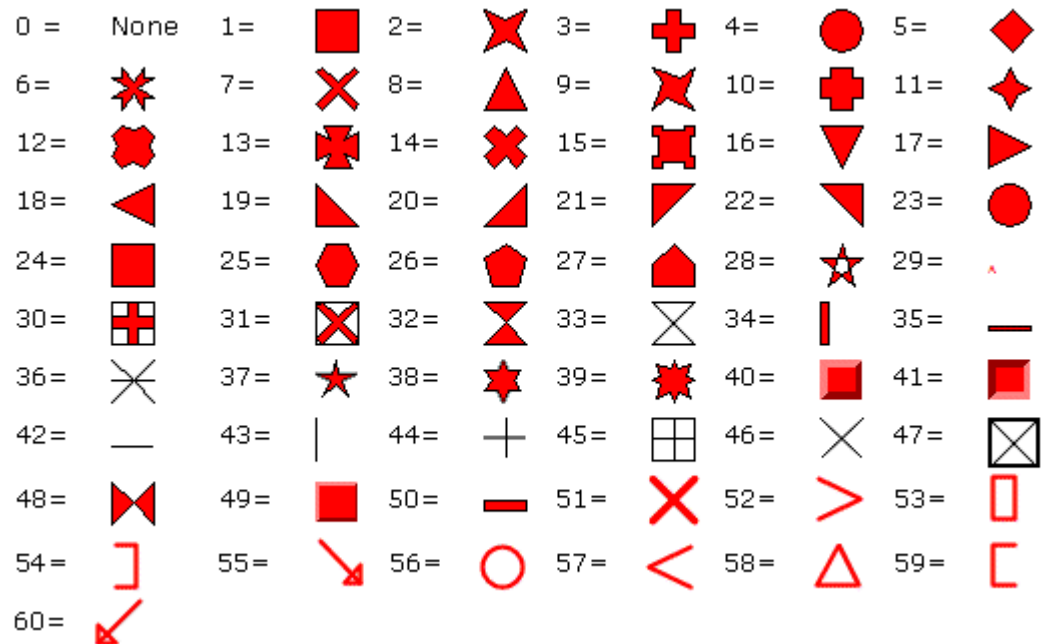
SYNTAX:

```
@MARKER nSeries nMarker
```

PARAMETERS:

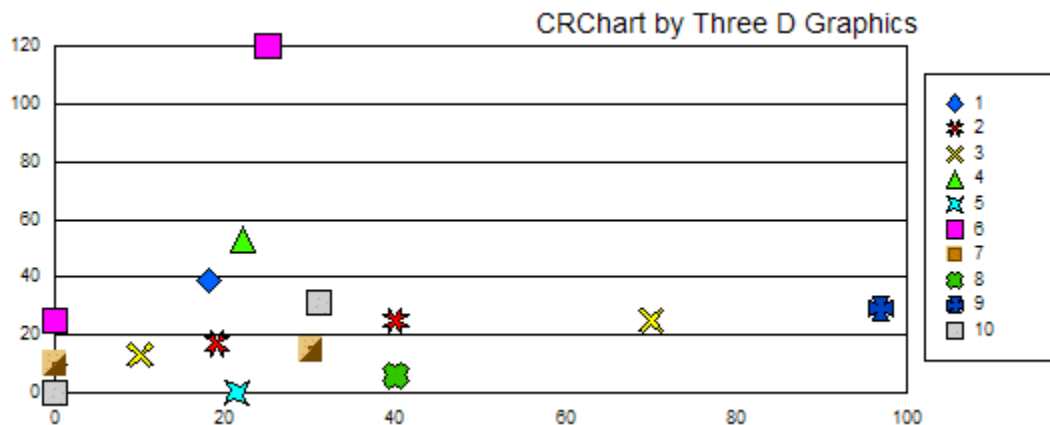
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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



EXAMPLE:

```
@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

@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.

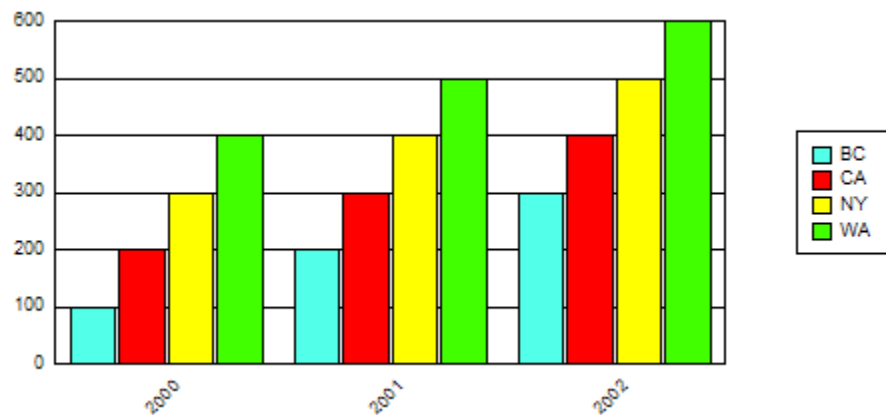
nR; 0...255 defines the Red portion of RGB color selection.

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.

@PAT (Riser Pattern)

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

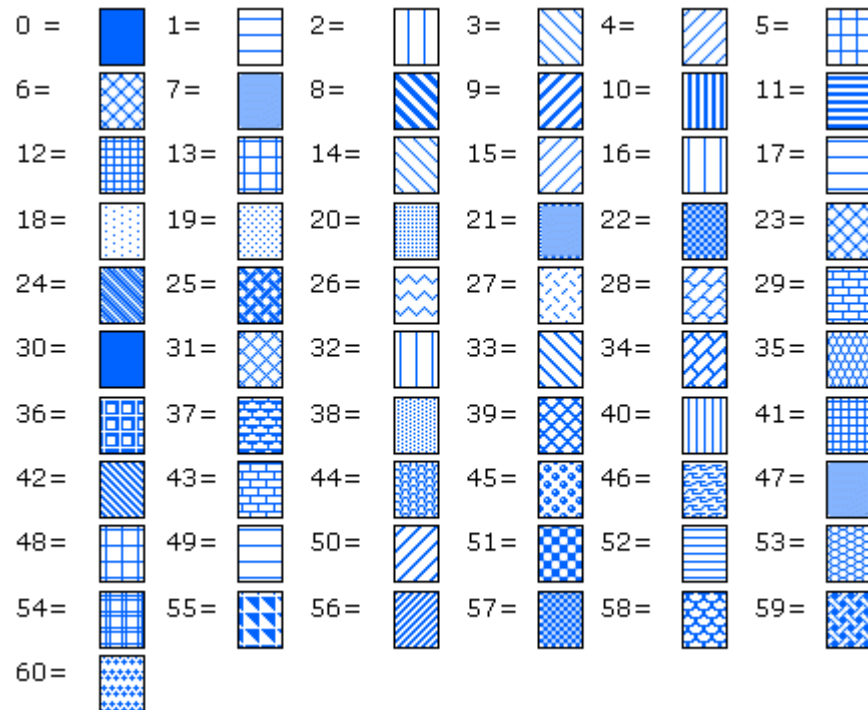
SYNTAX:

```
@PAT nSeries nPattern
```

PARAMETERS:

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

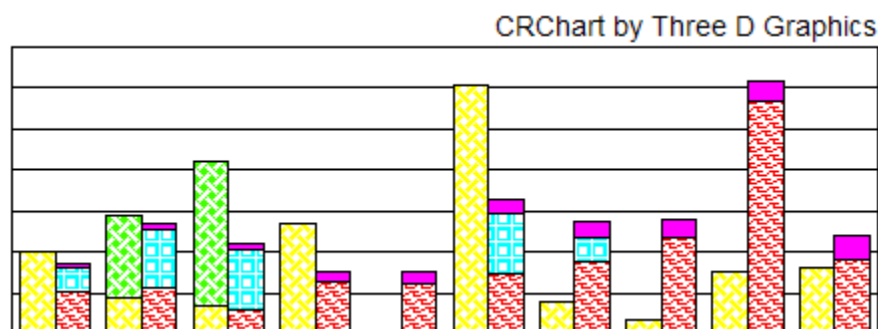
nPattern; -60...60. Use 0...60 to select one of the following patterns with a white background.



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

EXAMPLE:

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



PERSISTENT:

YES

@RISER_BORDER (Riser Border)

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

SYNTAX:

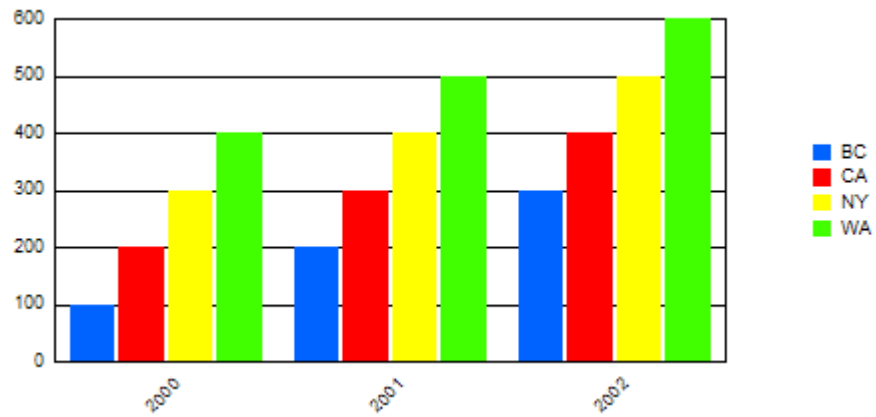
```
@RISER_BORDER bShow
```

PARAMETERS:

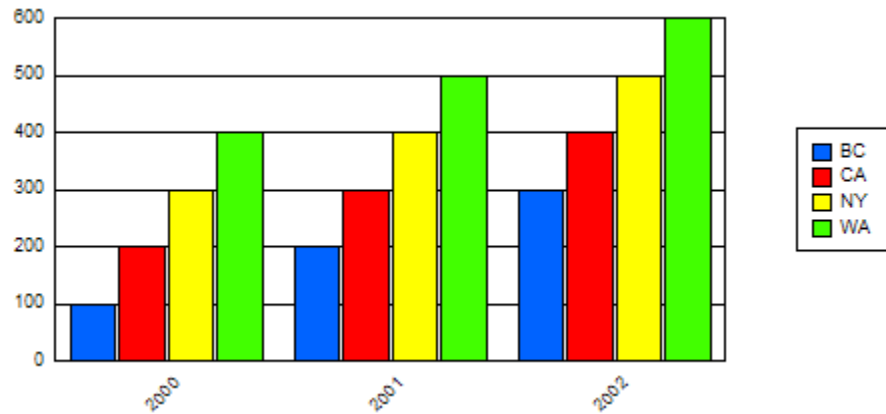
bShow; 0 = Turn OFF border line on risers/markers, 1 = Turn ON border line on risers/markers.

EXAMPLE:

```
@RISER_BORDER 0
```



```
@RISER_BORDER 1
```



PERSISTENT:

YES

@RISER_OVERLAP (Riser Overlap)

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

SYNTAX:

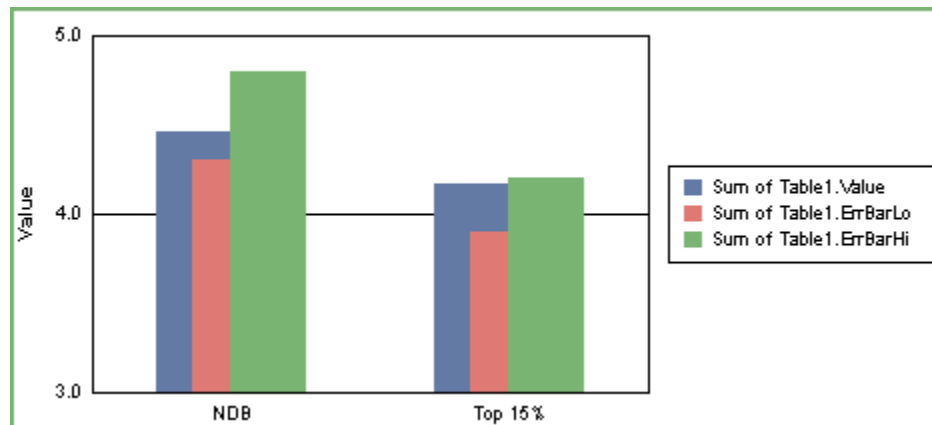
```
@RISER_OVERLAP n1
```

PARAMETERS:

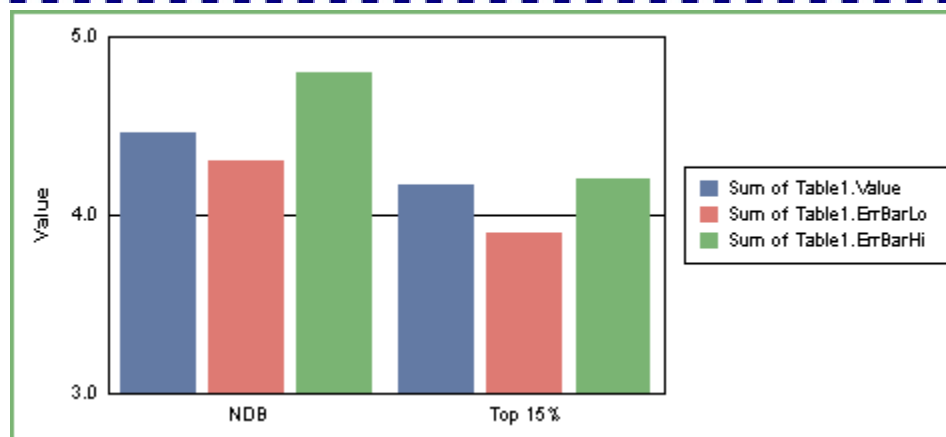
n1; -100...100 selects the amount of overlap. Negative numbers make the risers overlap. A value of -100 will place all risers in a group draw on top of each other. Larger numbers will draw the risers farther apart.

EXAMPLE:

```
@RISER_OVERLAP -50
```



```
@RISER_OVERLAP 50
```

**PERSISTENT:**

YES

@RISER_WIDTH (Riser Width)

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

SYNTAX:

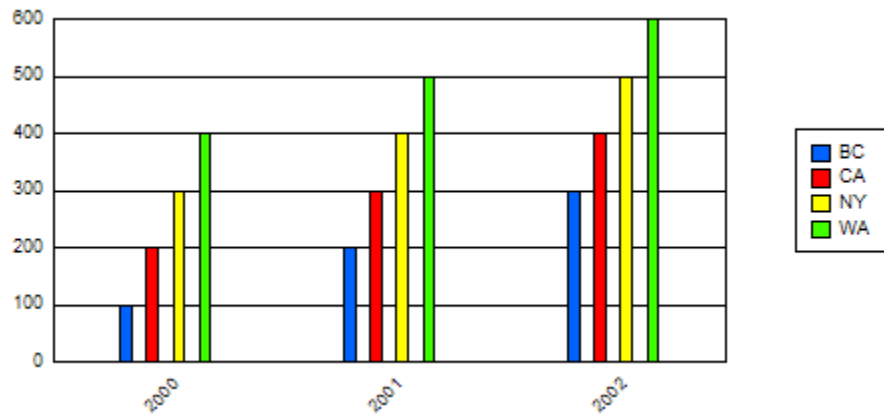
```
@RISER_WIDTH nWidth
```

PARAMETERS:

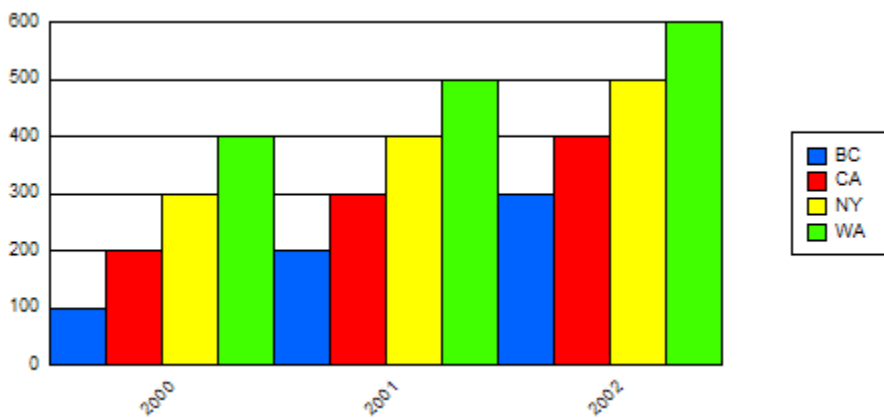
nWidth; 0...100 selects the width of risers.

EXAMPLE:

```
@RISER_WIDTH 20
```



```
@RISER_WIDTH 100
```



PERSISTENT:

YES



Section 7: Data & Data Text

Use these macros to manage data and data text:

- @APPEND_DATATEXT; Append a String to Data Text
- @DATATEXT; Select a Data Text Mode
- @DLT (Data Line Type)
- @DP; Override a Data Point
- @DPC; Clear a Data Point Clear
- @DT; Show/Hide Series-Specific Data Text
- @IR; Insert Row (i.e., create a user-defined series of data)
- @NEG_STYLE; Choose a style to indicate Negative Values
- @OFFSCALE_Y1; Define handling of Offscale Values on the Y1-Axis
- @RESET; Reset Internal Data Range
- @STRIP_ZERO; Strip Zero Values

@APPEND_DATATEXT (Append String to Data Text)

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

SYNTAX:

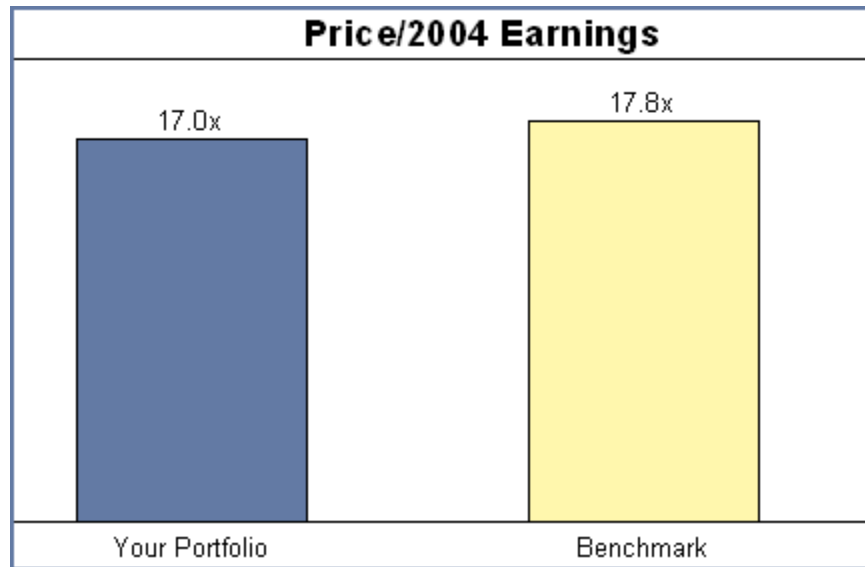
```
@APPEND_DATATEXT sz
```

PARAMETERS:

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

EXAMPLE:

```
@APPEND_DATATEXT x
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@DATATEXT (Data Text Mode)

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

SYNTAX:

```
@DATATEXT n
```

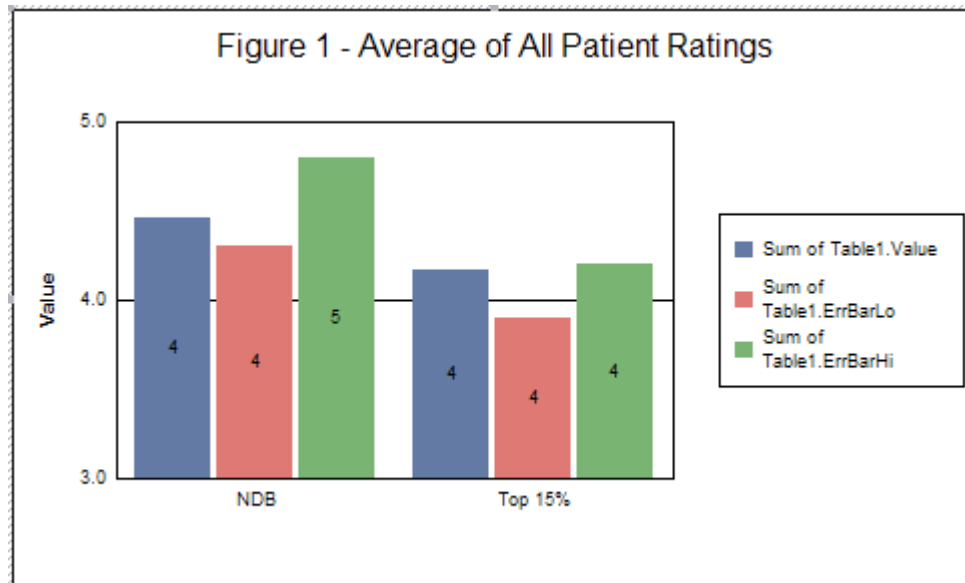
PARAMETERS:

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
```

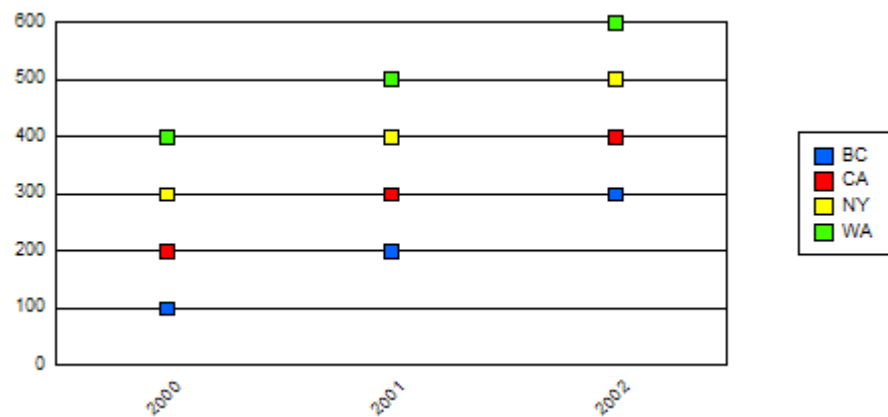
PARAMETERS:

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.

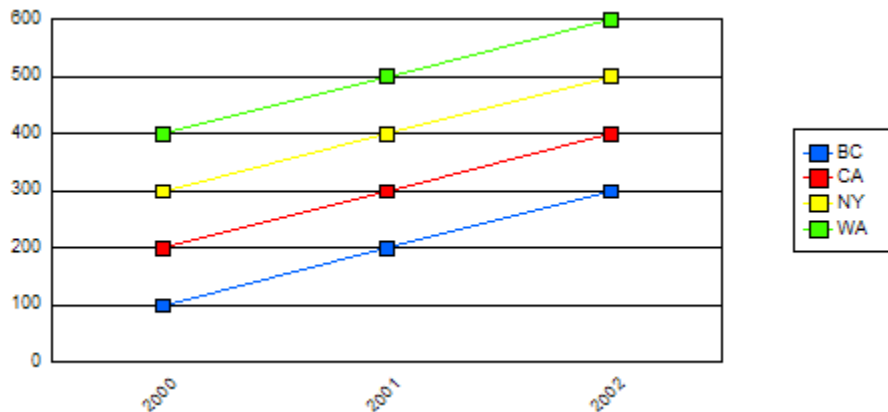
type; 1 = draw markers only, 2 = draw lines only, 3 = draw markers and lines.

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

@DP (Data Point Override)

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

SYNTAX:

```
@DP nSeries nGroup fValue
```

PARAMETERS:

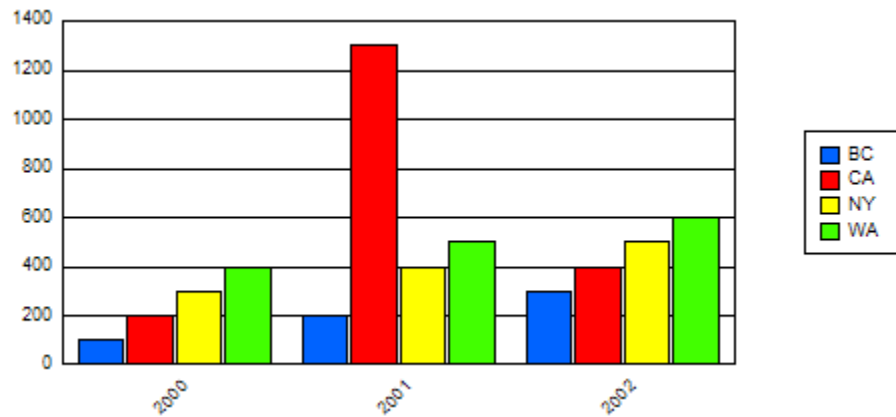
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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

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

EXAMPLE:

```
@DP 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
```

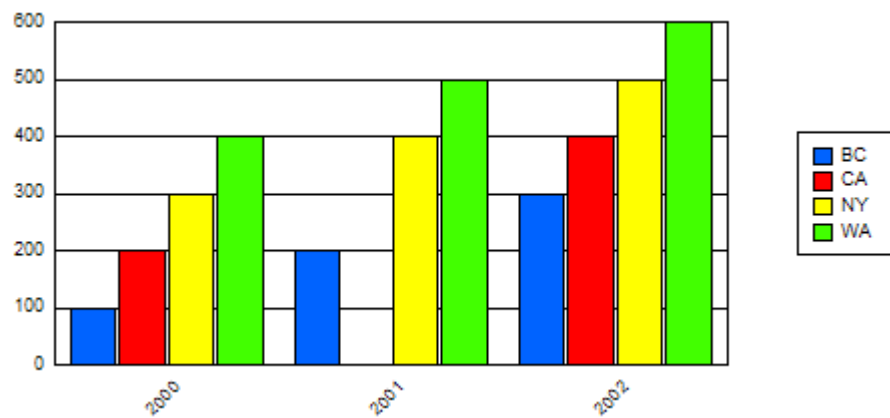
PARAMETERS:

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

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

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

SYNTAX:

```
@DT nSeries bShow
```

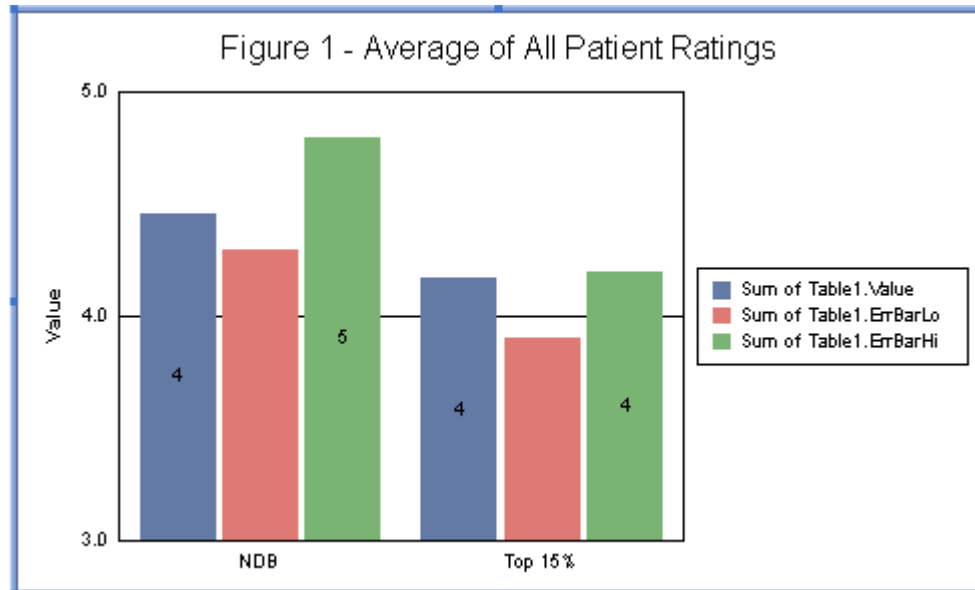
PARAMETERS:

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

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

EXAMPLE:

```
@DT 0 1
@DT 1 0
@DT 2 1
```



PERSISTENT:

YES

NOTES:

This macro requires Crystal Report 10.x or higher.

@NEG_STYLE (Negative Value Style)

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

SYNTAX:

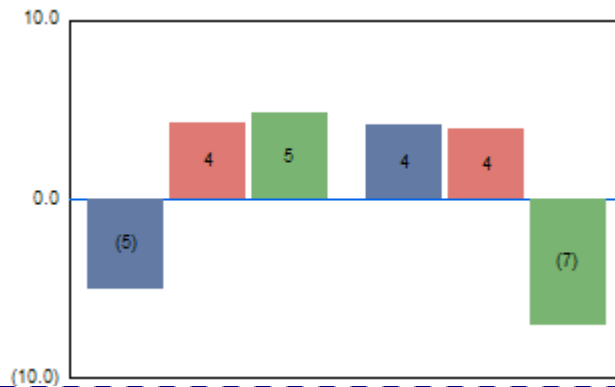
```
@NEG_STYLE nParenthesis
```

PARAMETERS:

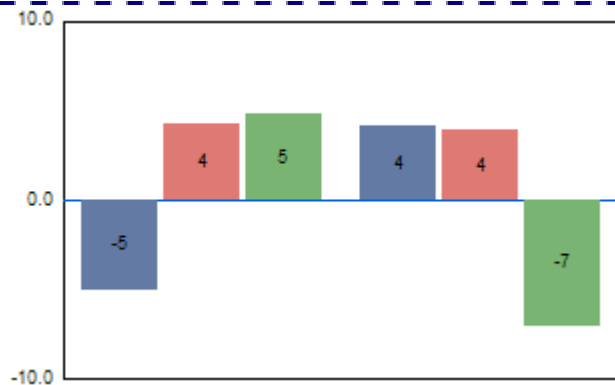
nParenthesis; 1 = format negative values with parenthesis (e.g., (1)), 0 = format negative values with a minus sign (e.g., -1)

EXAMPLE:

```
@NEG_STYLE 1
```



```
@NEG_STYLE 0
```



PERSISTENT:

YES

@OFFSCALE_Y1 (Y1-Axis Offscale Values)

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

SYNTAX:

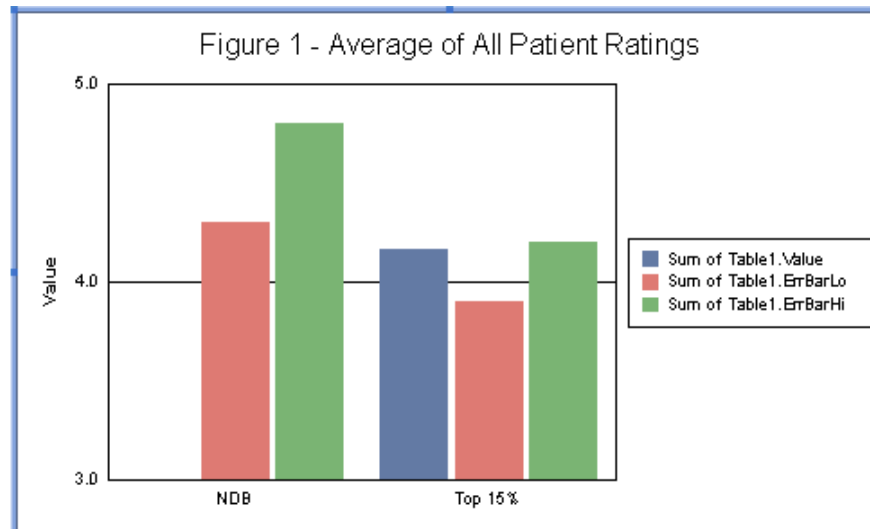
```
@OFFSCALE_Y1 nOffScaleMode
```

PARAMETERS:

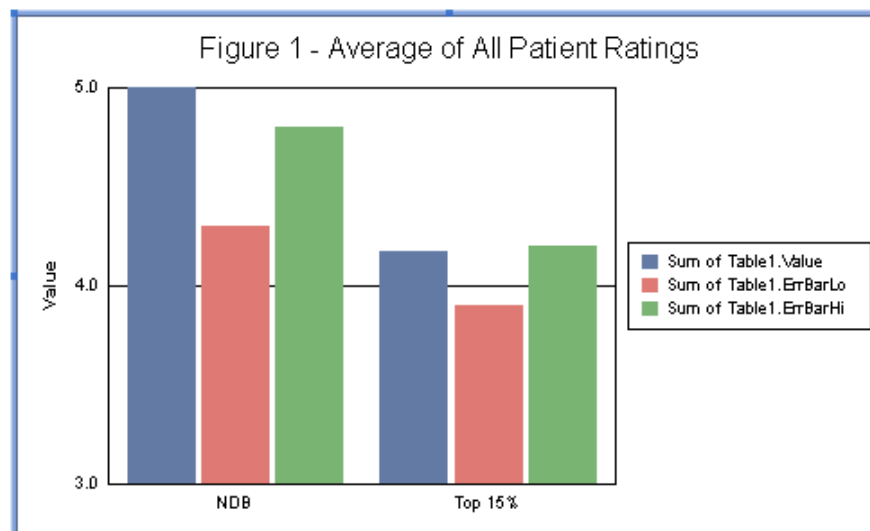
nOffScaleMode; 0...2. 0=Do not show offscale values, 1=Graph offscale values at scale maximum, 2=Clip offscale values against chart frame.

EXAMPLE:

```
@DP 0 0 500
@OFFSCALE_Y1 0 0
```



```
@OFFSCALE_Y1 0 1
```



PRESISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@RESET (Reset Internal Data Range)

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

SYNTAX:

@RESET

PARAMETERS:

None

PERSISTENT:

NO

@STRIP_ZERO (Strip Zero Values)

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

SYNTAX:

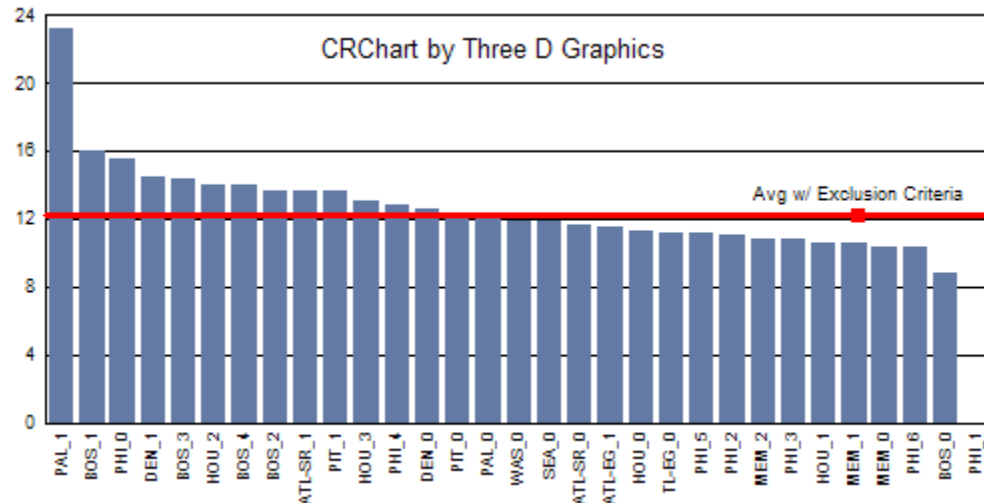
```
@STRIP_ZERO nSeries
```

PARAMETERS:

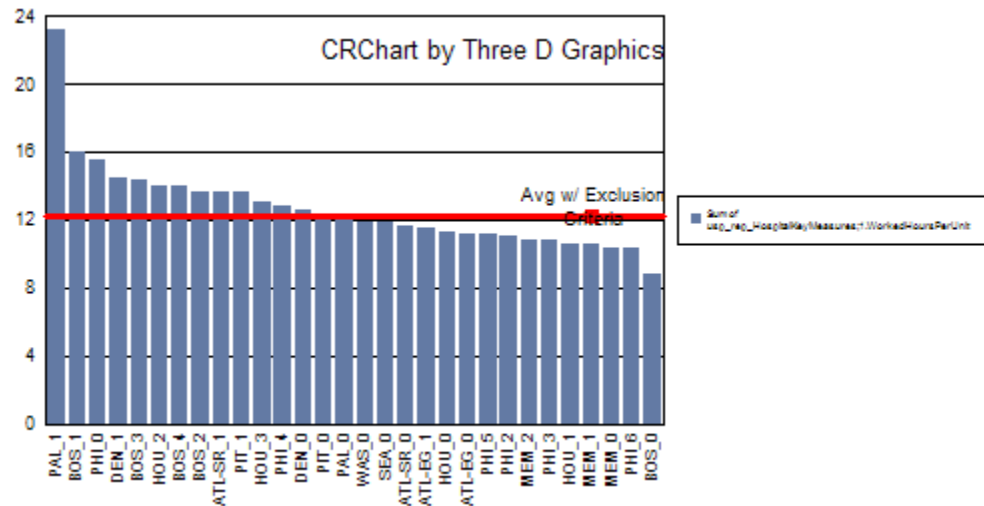
nSeries; Series Number (0...1024)

EXAMPLE:

Chart Before @STRIP_ZERO



@STRIP_ZERO 0



PERSISTENT:

YES



Section 8: Color & Special Effects

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

- @ALPHA; Apply Alpha Channel Transparency to a riser or marker
- @BEVEL; Draw a bevel effect on a chart object (**CHCHART PLUS**)
- @COLOR_MODE; Choose a Color Mode (Color by Series or Color by Group)
- @COND_COLOR & @COND_COLOR2; Apply color to risers (bar/line/area) based on conditions
- @CURVED_LINES; Apply curved lines to an line or area chart (**CHCHART PLUS**)
- @GCOLOR; Color a Chart Object
- @HQ; Enable/Disable high quality rendering (**CHCHART PLUS**)
- @SHADOW; Apply a Drop Shadow to a chart object

@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:

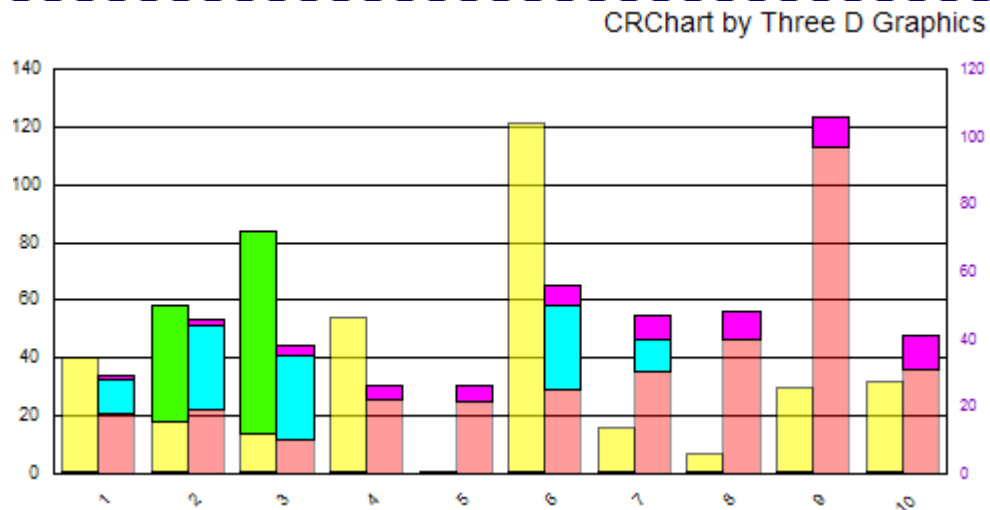
nSeries; -1...*n* (where: *n* = the total number of series in the chart). -1 = apply to all series, 0 = Series 1, 1 = Series 2, etc.

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

@BEVEL (Bevel Chart Object)

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

SYNTAX:

```
@BEVEL nObject nBevelType
```

PARAMETERS:

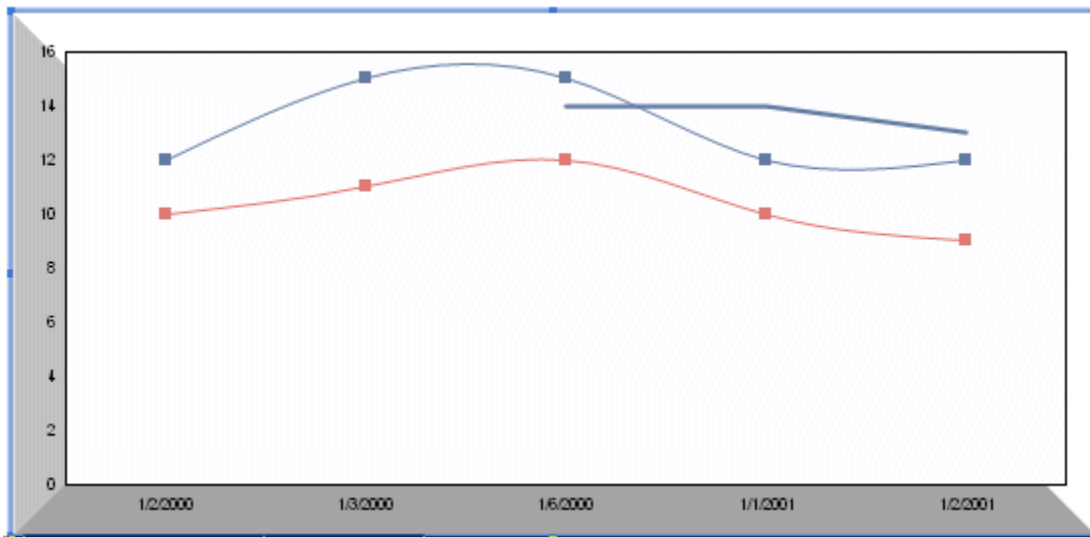
nObject; 1...3. 1=Chart Background, 2=Chart Frame. 3=Risers

nBevelType; 0...3. 0=None, 1=Smooth Edge, 2=Chisel Edge, 3=Donut.

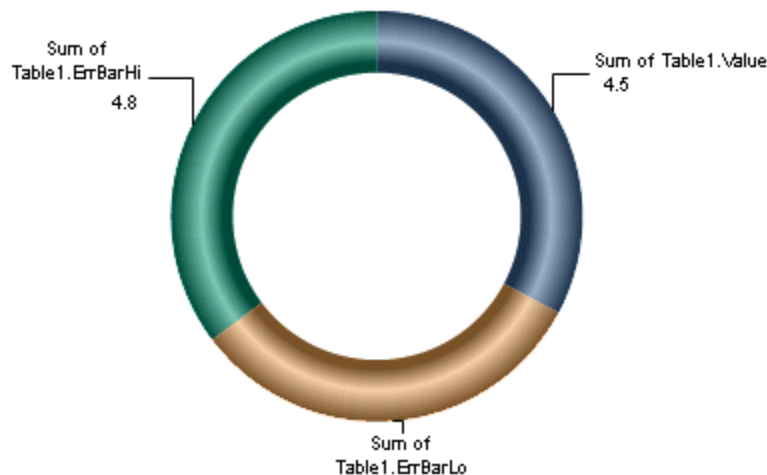
Bevel types 1 and 2 can only be applied to rectangle and polygons objects. Bevel type 3 can only be applied to ellipses and slices.

EXAMPLE:

```
@BEVEL 1 1
```

**EXAMPLE:**

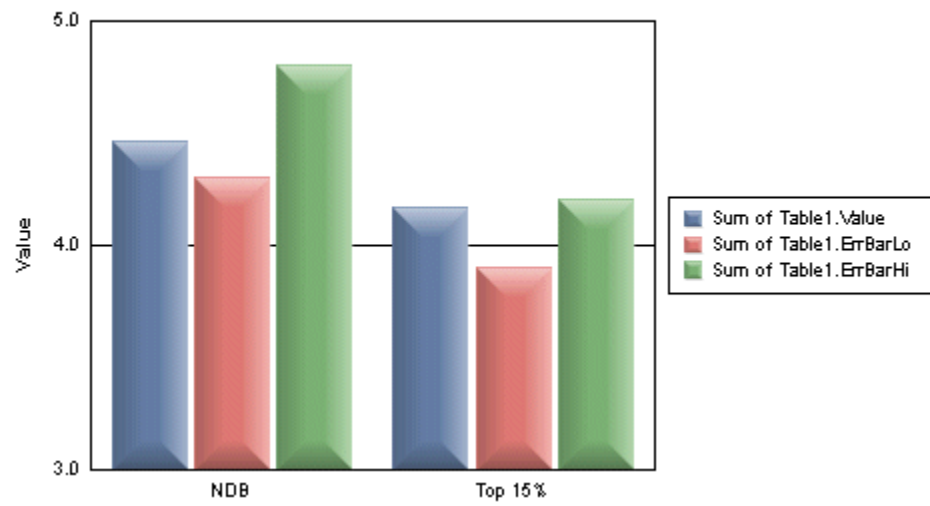
```
@BEVEL 3 3
```



EXAMPLE:

@BEVEL 3 1

Figure 1 - Average of All Patient Ratings



PRESISTENT:

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

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@COLOR_MODE (Color Mode)

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

SYNTAX:

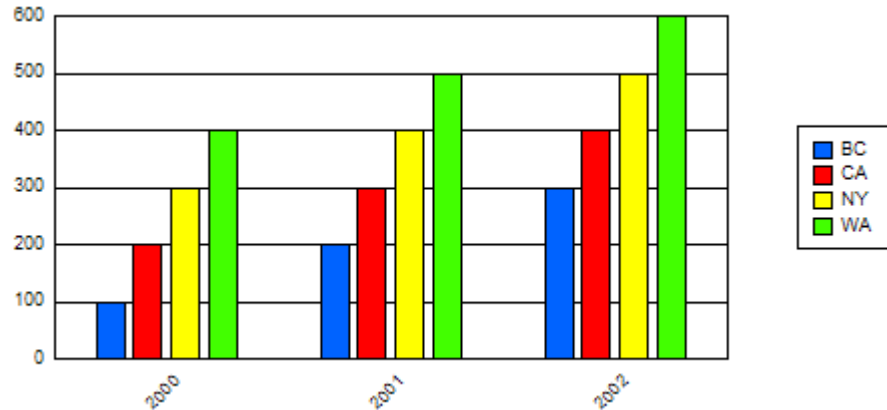
```
@COLOR_MODE nMode
```

PARAMETERS:

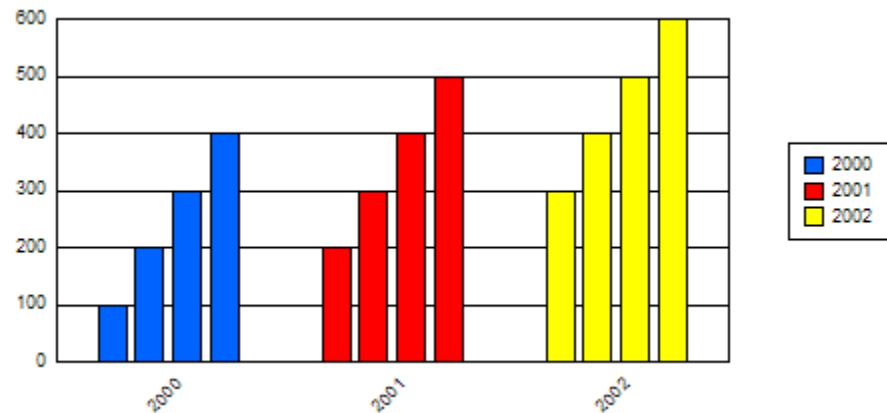
nMode; 0 = "Color By Series", 1 = "Color By Group"

EXAMPLE:

```
@COLOR_MODE 0
```



```
@COLOR_MODE 1
```



PERSISTENT:

NO

@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.

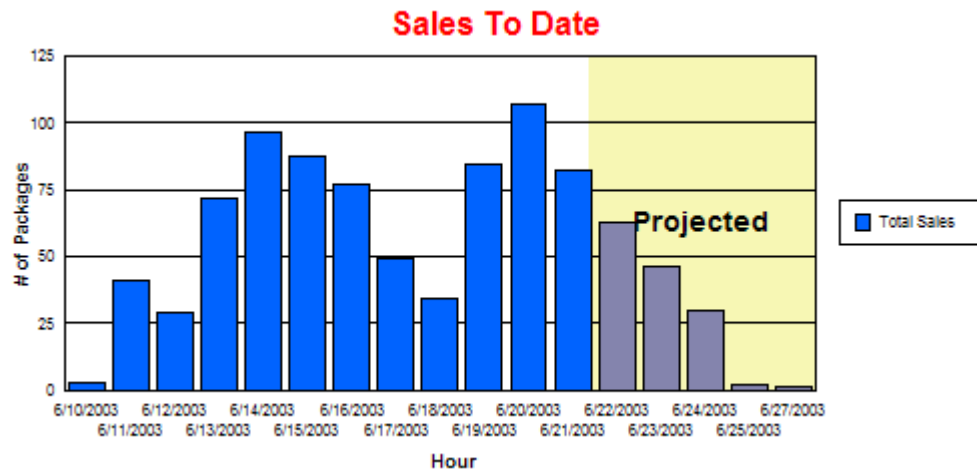
7 = force all series in *nGroup* to the conditional color regardless of *fValue*.

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.

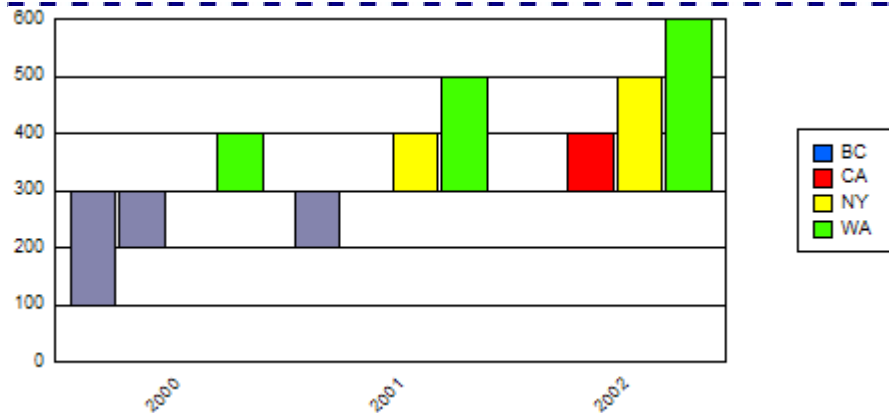
Examples:

```
@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`).

@CURVED_LINES (Enable Curved Lines)



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

SYNTAX:

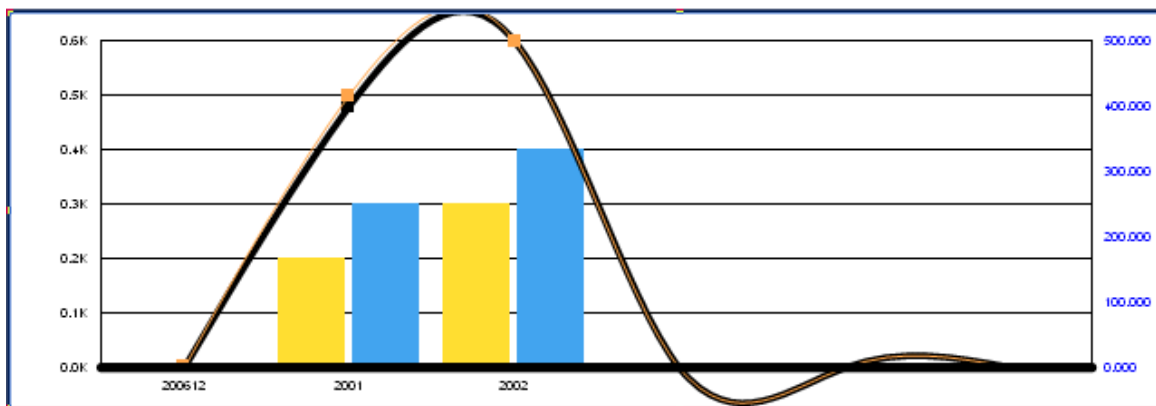
```
@CURVED_LINES bEnable
```

PARAMETERS:

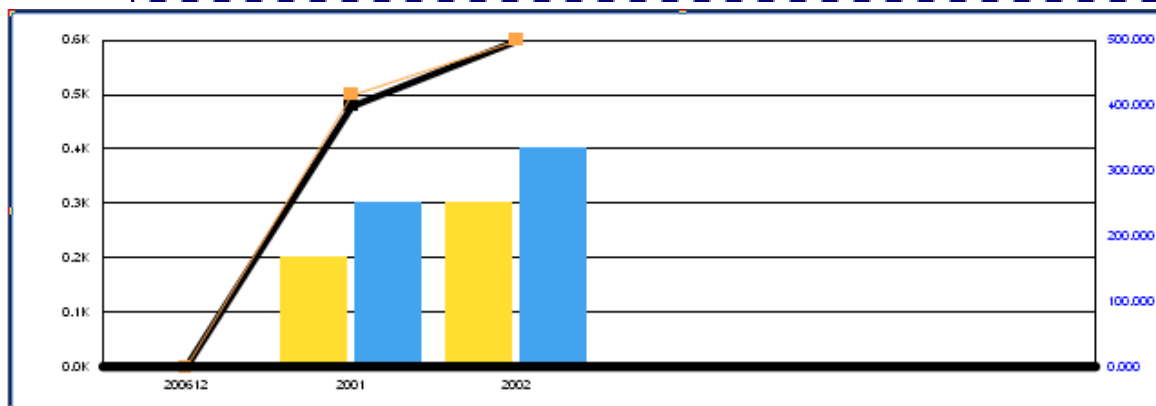
bEnable; 0=Disable Curved Lines, 1=Enable Curved Lines

EXAMPLE:

```
@CURVED_LINES 1
```



```
@CURVED_LINES 0
```



PRESISTENT:

YES

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GCOLOR (Color Chart Object)

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

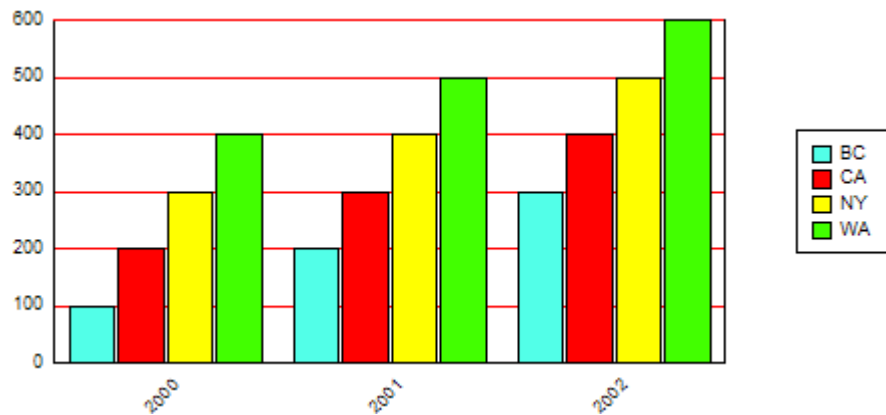
nR: 0...255 defines the Red portion of RGB color selection.

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

@HQ (Enable High-Quality Rendering)



This macro enables/disables high-quality rendering. High-Quality rendering will apply anti-aliasing to all chart elements in Crystal Reports 9, 10 and 11 and greatly improve the overall appearance of these charts.

SYNTAX:

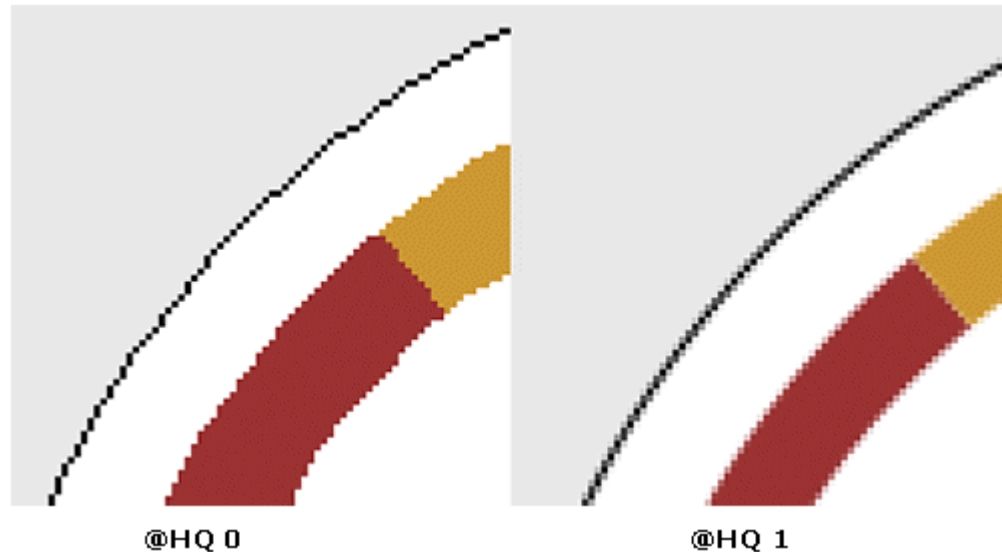
@HQ bEnable

PARAMETERS:

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

EXAMPLE:

Close-Up View of High-Quality Rendering



PRESISTENT:

NO

NOTES:

- This macro requires CRChart Plus.
- @HQ is only useful in Crystal Reports 9, 10 and 11. Crystal Reports 11.5, Release 2 applies anti-aliasing to most chart elements.
- This macro will automatically be enabled (@HQ 1) when another macro requires high-quality drawing. The following macros require high-quality rendering: @BEVEL, @CURVED_LINES, @GAUGE_BORDER_STYLE, @GAUGE_BORDER_THICKNESS, @GAUGE_COLOR, @GAUGE_MULTIPLE_NEEDLES, @GAUGE_NEEDLE_STYLE, @GAUGE_STYLE.

@SHADOW (Drop Shadow)

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

SYNTAX:

```
@SHADOW nObject nXOffset nYOffset
```

PARAMETERS:

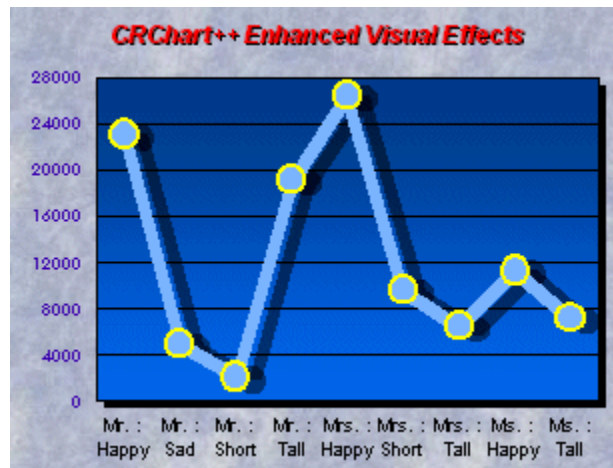
nObject; 0...12 selects one of the following objects:

- 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



Section 9: Lines

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

- @CXY; Draw a line between two X/Y Coordinates with Color
- @CY; Draw a horizontal line on the Y1-Axis with Color
- @LS; Select a Line Style
- @MEAN; Enable/disable a mean average line across a specified series
- @MOVA; Apply a standard or scientific Moving Average Line to a series
- @SMOOTH_LINE; Connect data points with Smooth or Straight Line segments
- @TRENDLINE; Draw a Trend Line across a series
- @TRENDLINE_ALLDATA; Draw a Linear Regression Line through all data points
- @X; Draw a vertical line on the X-Axis at Value
- @XG; Draw a vertical line on the X-Axis at Group
- @XY; Draw a line between two X/Y Coordinates
- @XY_DP2; Draw a lines between two Data Points (Scatter Charts)
- @Y; Draw a horizontal line on the Y1-Axis
- @YSZ; Draw a horizontal line on the Y1-Axis with Label
- @YSZL; Draw a horizontal line on the Y1-Axis with Label on Left
- @YSZN; Draw a horizontal line on the Y1-Axis with Label & Value
- @YSZNL; Draw a horizontal line on the Y1-Axis with Label & Value on Left

@CXY (X/Y Coordinates Line with Color)

This macro adds a user-defined line that starts at location *x1*, *y1* and stops at location *x2*, *y2*. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *x1* and *x2* define the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *x1* and *x2* must be set to a value in the range 0.0 to 1.0 that defines a percentage of the X (or ordinal)-Axis length. Use the *nRed*, *nGreen*, and *nBlue* parameters to specify the RGB color of the line.

SYNTAX:

```
@CXY x1 y1 x2 y2 nRed nGreen nBlue
```

PARAMETERS:

x1; Beginning x-coordinate

y1; Beginning y-coordinate

x2; Ending x-coordinate

y2; Ending y-coordinate

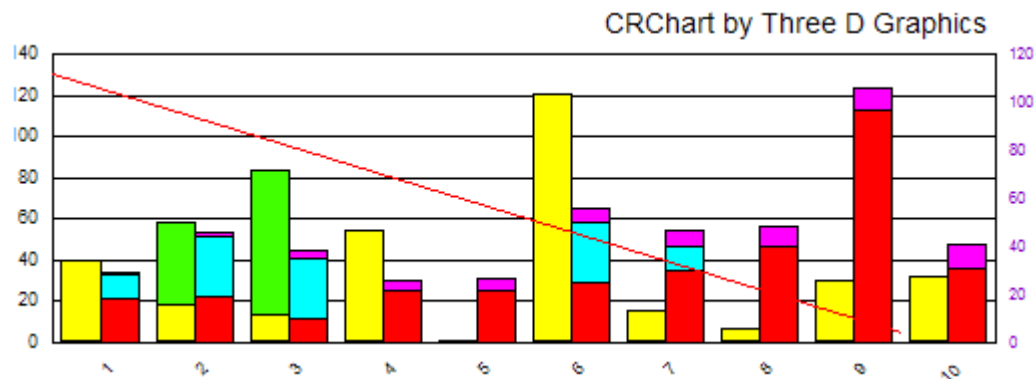
nRed; 0...255

nGreen; 0...255

nBlue; 0...255

EXAMPLE:

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



PERSISTENT:

NO

@CY (Y-Axis Horizontal Line with Color)

This macro adds a user-defined line on the y-axis at value n . Use the $nRed$, $nGreen$, and $nBlue$ parameters to specify the RGB color of the line.

SYNTAX:

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

PARAMETERS:

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

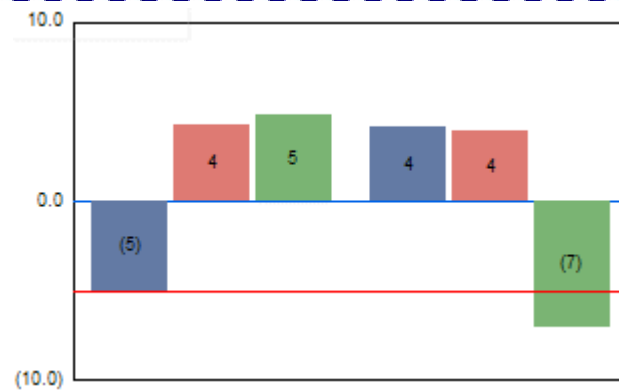
$nRed$; 0...255

$nGreen$; 0...255

$nBlue$; 0...255

EXAMPLE:

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



PERSISTENT:

NO

@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.

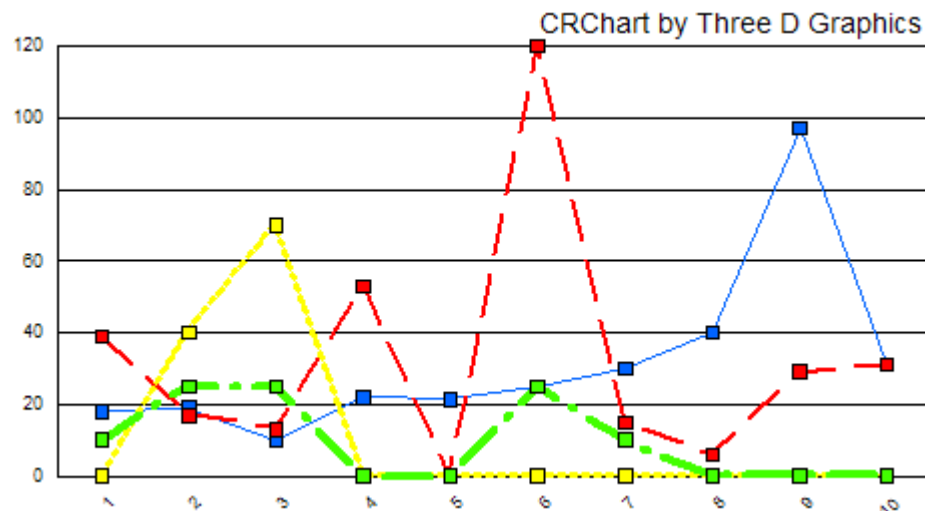
nThickness; 0...1000 selects the thickness of the line.

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

@MEAN (Mean Line)

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

SYNTAX:

```
@MEAN nSeries bShow
```

PARAMETERS:

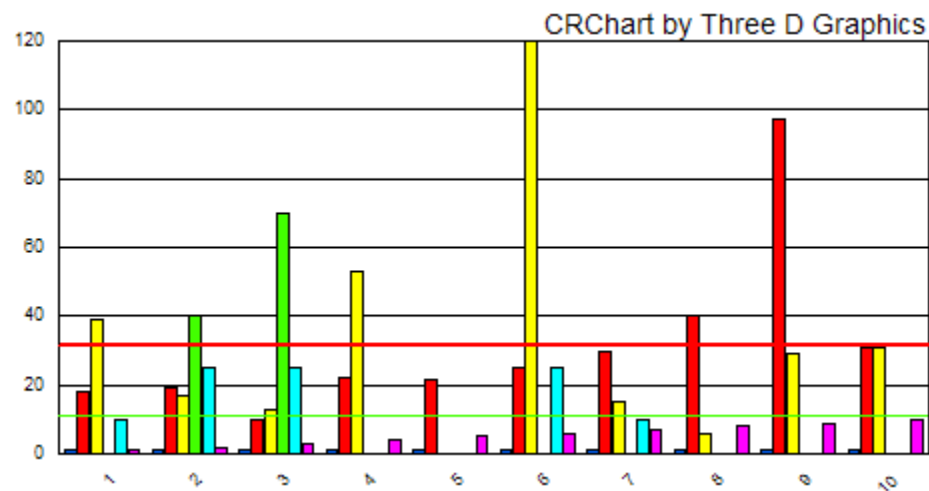
nSeries; 0...number of series in chart (0=Series 1).

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

@MOVA (Moving Average Line)

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

SYNTAX:

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

PARAMETERS:

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

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

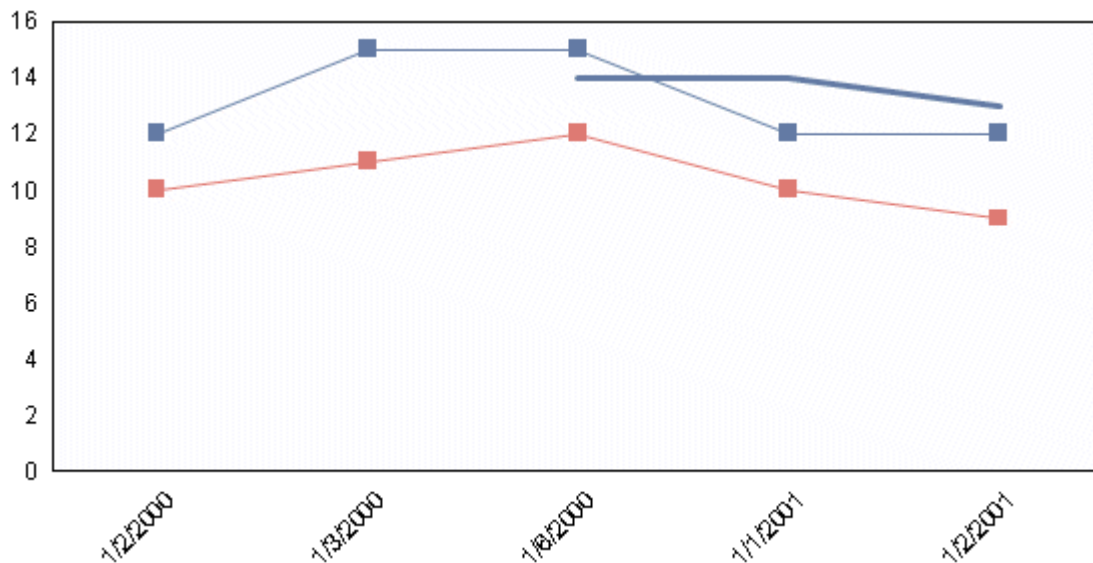
bSciMovMode; 1 = Scientific moving average. 0 = Standard moving average

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

EXAMPLE:

This example sets a normal 3-day moving average to the first series.

```
@MOVA 0 1 0 3
```

**PERSISTENT:**

YES

ALSO SEE:

@TRENDLINE to draw a trend line in a chart

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

NOTES:

This macro requires Crystal Reports 10.x or higher.

@SMOOTH_LINE (Smooth/Straight Lines)

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

SYNTAX:

```
@SMOOTH_LINE nSeries bOnOff
```

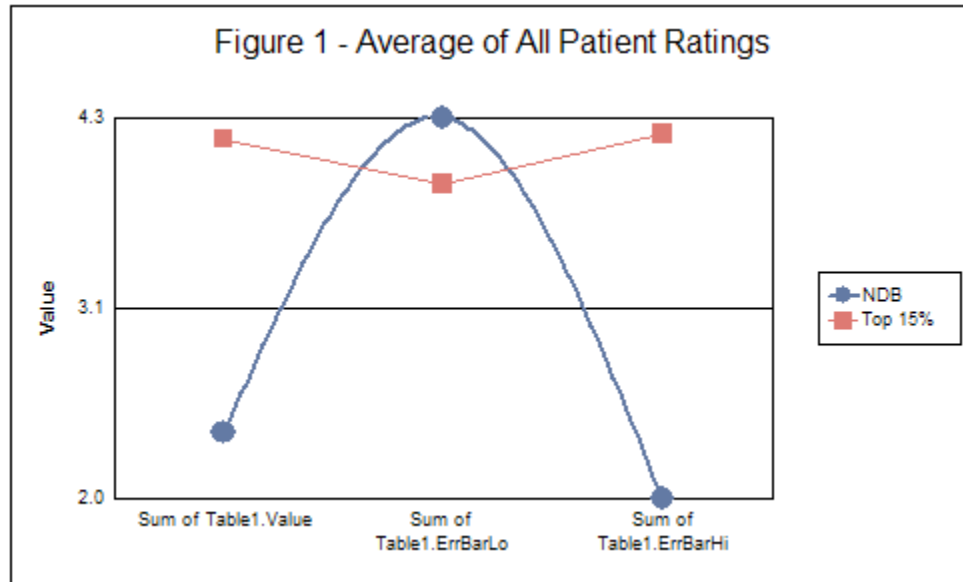
PARAMETERS:

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

@TRENDLINE (Trend Line)

This macro draws a trend line across a specified series.

SYNTAX:

```
@TRENDLINE nSeries nValue
```

PARAMETERS:

nSeries; Series (0...1000) to apply the trend line.

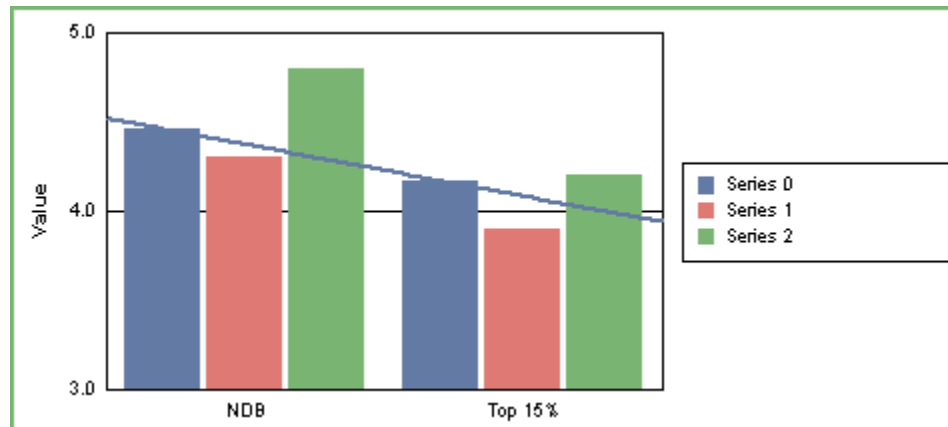
nValue; Bitwise flags (0...127) to activate one or more of the following trend lines:

- 1 = MEAN
- 2 = Standard Deviation
- 4 = Linear Regression
- 8 = Natural Log Regression
- 16 = Polynomial Regression
- 32 = Exponential Regression
- 64 = Log Regression

EXAMPLE:

This example draws both Linear AND Polynomial regression lines on the first series:

```
@TRENDLINE 0 20
```



PRESISTENT:

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.

ALSO SEE:

@MOVA to draw a moving average line in a chart

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

@TRENDLINE_ALLDATA (Linear Regression Line)

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

SYNTAX:

```
@TRENDLINE_ALLDATA bShowLinearLine bShowEquation
```

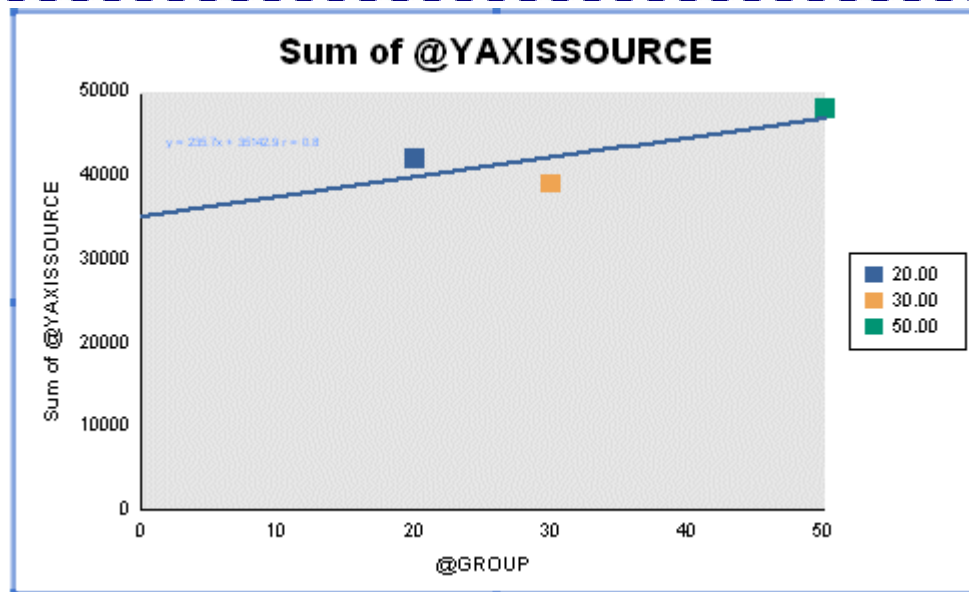
PARAMETERS:

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

bShowEquation; 1=Show Equation, 1=High Equation

EXAMPLE:

```
@TRENDLINE_ALLDATA 1 1
```



PRESISTENT:

YES

NOTES:

This macro requires Crystal Reports 10.x or higher.

ALSO SEE:

@MOVA to draw a moving average line in a chart

@TRENDLINE to draw a trend line

@X (X-Axis 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:

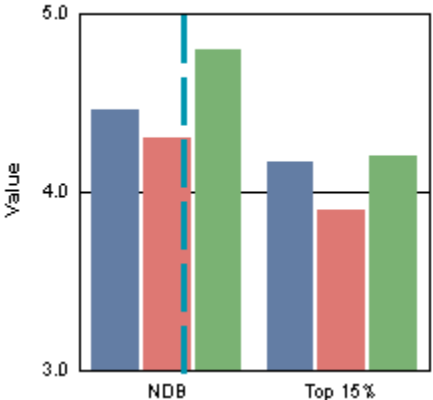
n; Value at which to add the user-defined line on the x-axis

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

@XG (X-Axis Vertical Line at Group)

This macro is like the @X macro except it draws a line at a specified group. It allows you place the vertical line more accurately (using *groupID*) on Bar/line/area/BoxPlots.

SYNTAX:

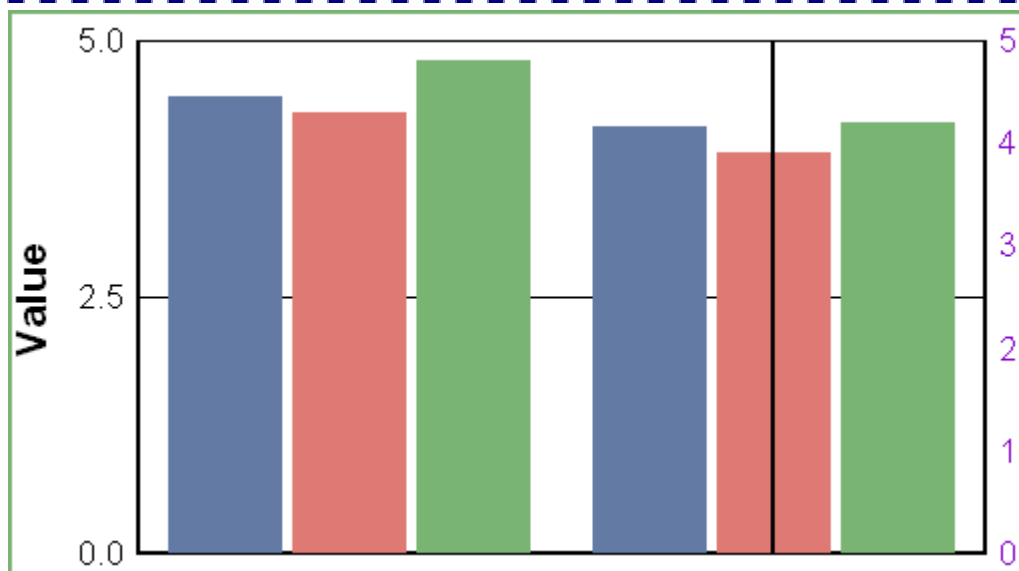
```
@XG groupID
```

PARAMETERS:

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

EXAMPLE:

```
@XG 1
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@XY (X/Y Coordinates Line)

This macro adds a user-defined line that starts at location *nx1*, *ny1* and stops at location *nx2*, *ny2*. For charts with a true X-Axis (e.g., Scatter, Bubble, Polar, etc.), *nx1* and *nx2* define the value on the X-Axis where the line will be drawn. For bar, line, or area charts, *nx1* and *nx2* must be set to a value in the range 0.0 to 1.0 that defines a percentage of the X (or ordinal)-Axis length.

SYNTAX:

```
@XY nx1 ny1 nx2 ny2
```

PARAMETERS:

nx1; Beginning x-coordinate

ny1; Beginning y-coordinate

nx2; Ending x-coordinate

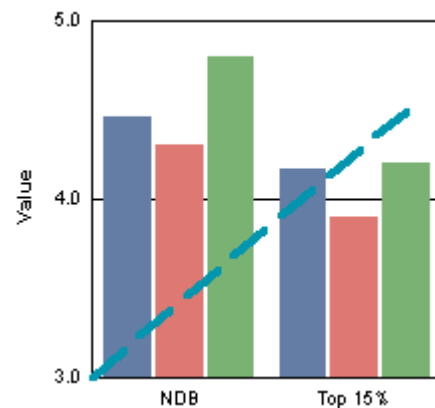
ny2; Ending y-coordinate

EXAMPLE:

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

@Y (Y1-Axis Horizontal Line)

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

SYNTAX:

```
@Y n
```

PARAMETERS:

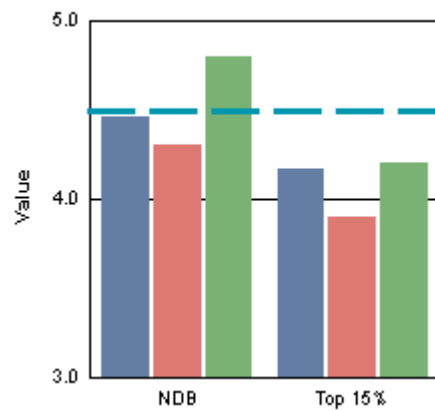
n ; Value at which to add the user-defined line on the y-axis

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

@YSZ (Y-Axis Line with Label)

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:

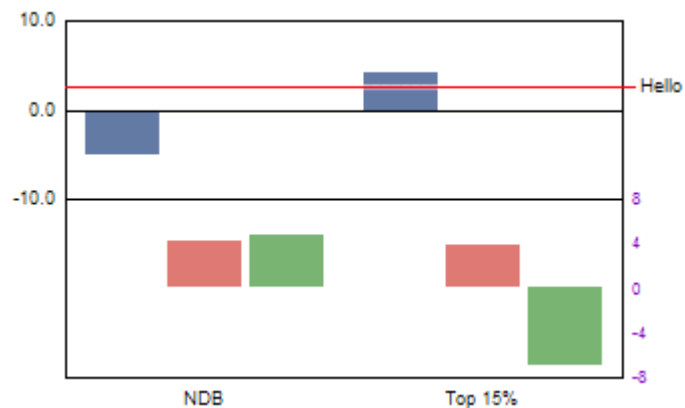
fValue; Value at which to add the user-defined line on the y-axis

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. Add a tilde (~) character to the end of the string if you intend to define other CRChart macros in the same title field.

EXAMPLE:

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



PERSISTENT:

NO

ALSO SEE:

@YSZL to draw the label on the left side of the chart.

@YSZN to show *fValue* with the label string on the right side of the chart.

@YSZNL to show *fValue* with the label string on the left side of the chart.

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

This macro adds a user-defined line on the y-axis at value *fValue* with the label *sZLabel*. The label is drawn on the left side of the chart.

SYNTAX:

```
@YSZL fValue nXFixup nYFixup sZLabel
```

PARAMETERS:

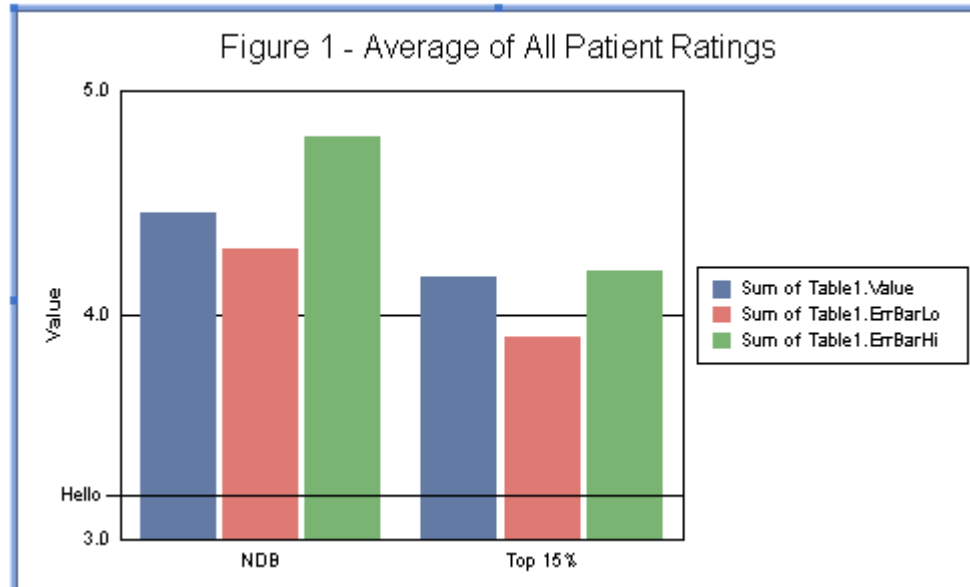
fValue; Value at which to add the user-defined line on the y-axis

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. Add a tilde (~) character to the end of the string if you intend to define other CRChart macros in the same title field.

EXAMPLE:

```
@YSZL 3.2 0 0 Hello
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

ALSO SEE:

@YSZ to draw the label on the right side of the chart.

@YSZN to show *fValue* with the label string on the right side of the chart.

@YSZNL to show *fValue* with the label string on the left side of the chart.

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

This macro adds a user-defined line on the Y-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*.

SYNTAX:

```
@YSZN fValue nXFixup nYFixup sZLabel nValue
```

PARAMETERS:

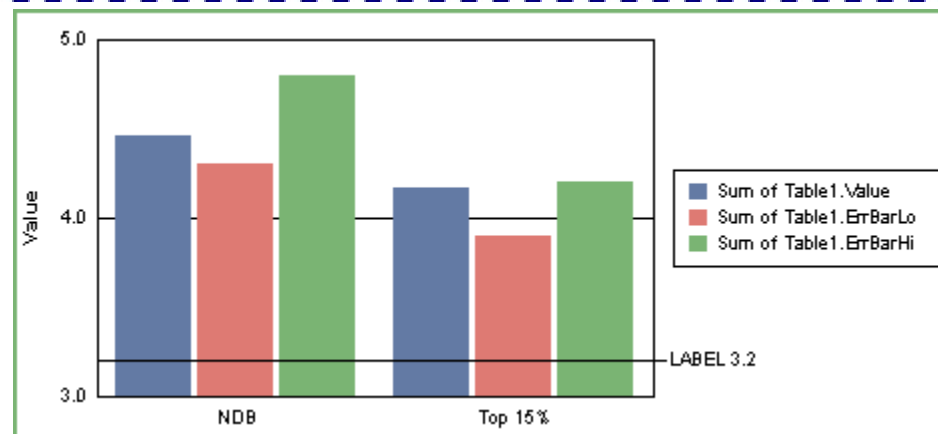
fValue; Value at which to add the user-defined line on the y-axis

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. Add a space to the end of your label if you want a space to appear between the label and *fValue*.

EXAMPLE:

```
@YSZN 3.2 0 0 LABEL
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

ALSO SEE:

@YSZNL to draw the label and value on the left side of the chart.

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

This macro adds a user-defined line on the Y-axis at value *fValue* with the label *sZLabel*. The value of *fValue* is appended to *sZLabel*. The label and value are drawn on the left side of the chart.

SYNTAX:

```
@YSZNL fValue nXFixup nYFixup sZLabel nValue
```

PARAMETERS:

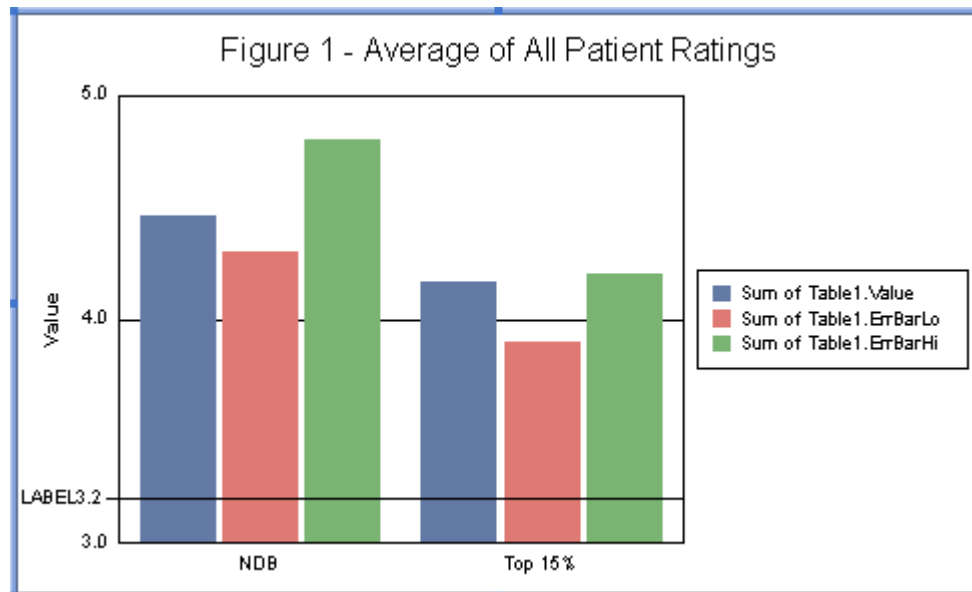
fValue; Value at which to add the user-defined line on the Y-axis

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. Add a space to the end of your label if you want a space to appear between the label and *fValue*.

EXAMPLE:

```
@YSZNL 3.2 0 0 LABEL
```

**PERSISTENT:**

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

ALSO SEE:

@YSZN to draw the label and value on the right side of the chart.



Section 10: User-Defined Objects

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

- @USER_CIRCLE; Draw a User-Defined Outlined Circle
- @USER_CIRCLE_ABOVE; Draw a User-Defined Outlined Circle above the Chart Area
- @USER_FILL; Draw a User-Defined Color-Filled Rectangle
- @USER_FILL_CIRCLE; Draw a User-Defined Color-Filled Circle
- @USER_FILL_CIRCLE_ABOVE; Draw a User-Defined Color-Filled Circle above the Chart Area
- @USER_FILL_CIRCLE2; Draw a User-Defined Pattern-Filled Circle
- @USER_FILL_CIRCLE2_ABOVE; Draw a User-Defined Pattern-Filled Circle Above the Chart Area
- @USER_FILL2; Draw a User-Defined Pattern-Filled Rectangle
- @USER_MARKER; Draw a User-Defined Marker at X/Y coordinates
- @USER_MARKER2; Draw a User-Defined Marker at X/Y coordinates with Value
- @USER_RECT; Draw a User-Defined Outlined Rectangle

@USER_CIRCLE (Outlined Circle)

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

SYNTAX:

```
@USER_CIRCLE fStartX fStopX fStartY fStopY nRed nGreen nBlue
nThickness nStyle 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

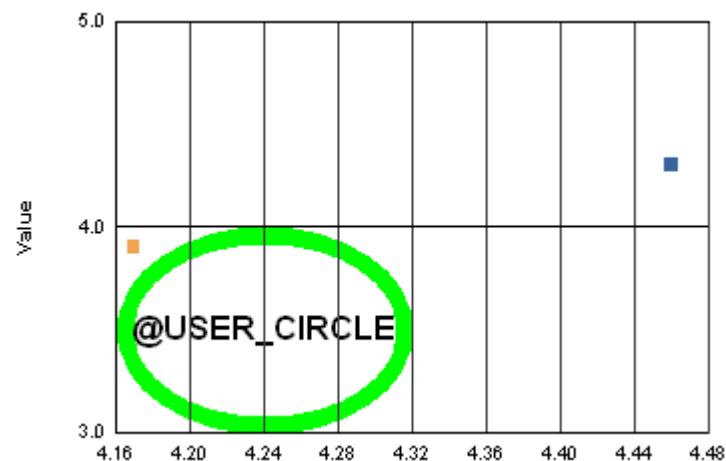
nThickness; 0...1000 selects the thickness of the outline.

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

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

EXAMPLE:

```
@USER_CIRCLE 0.0 0.5 0.0 0.5 0 255 0 0 1000 @USER_CIRCLE~
```

**PERSISTENT:**

NO

NOTES:

- A maximum of 5 @USER_CIRCLE, @USER_FILL..., and/or @USER_RECT macros may be specified but any combination of the macros may be used. @USER_CIRCLE, @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.
- This macro requires Crystal Reports 9.x or higher.

@USER_CIRCLE_ABOVE (Outlined Circle above Chart Area)

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

SYNTAX:

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

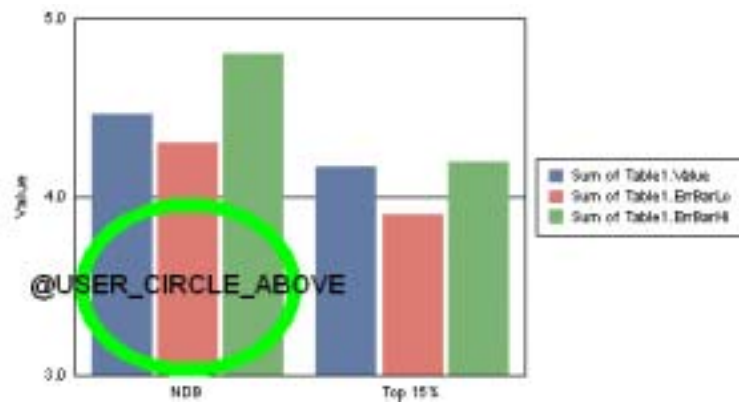
nThickness; 0...1000 selects the thickness of the outline.

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

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

EXAMPLE:

```
@USER_CIRCLE_ABOVE 0.0 0.5 0.0 0.5 0 255 0 0 1000
@USER_CIRCLE_ABOVE~
```



PERSISTENT:

NO

NOTES:

- A maximum of 5 @USER_CIRCLE, @USER_FILL..., and/or @USER_RECT macros may be specified but any combination of the macros may be used. @USER_CIRCLE, @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.
- This macro requires Crystal Reports 9.x or higher.

@USER_FILL (Color-Filled Rectangle)

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

SYNTAX:

```
@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

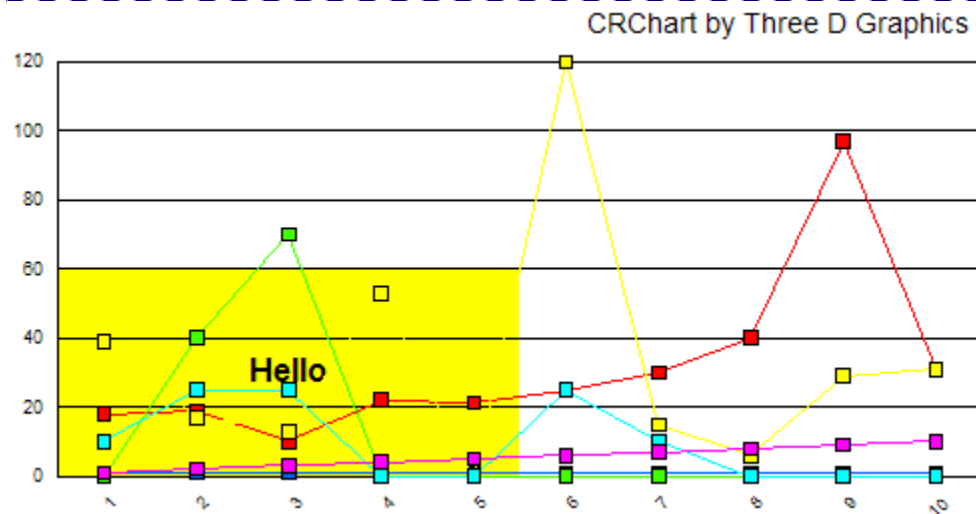
nRed; 0...255, *nGreen*; 0...255, *nBlue*; 0...255

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

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" in the middle.

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



PERSISTENT:

NO

NOTES:

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

@USER_FILL2 (Pattern-Filled Rectangle)

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

SYNTAX:

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

PARAMETERS:

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

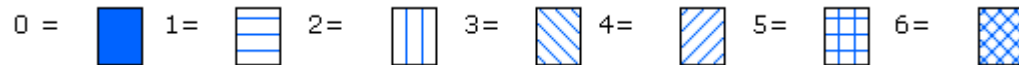
fStopX; 0.0...1.1 X-Axis stop location

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

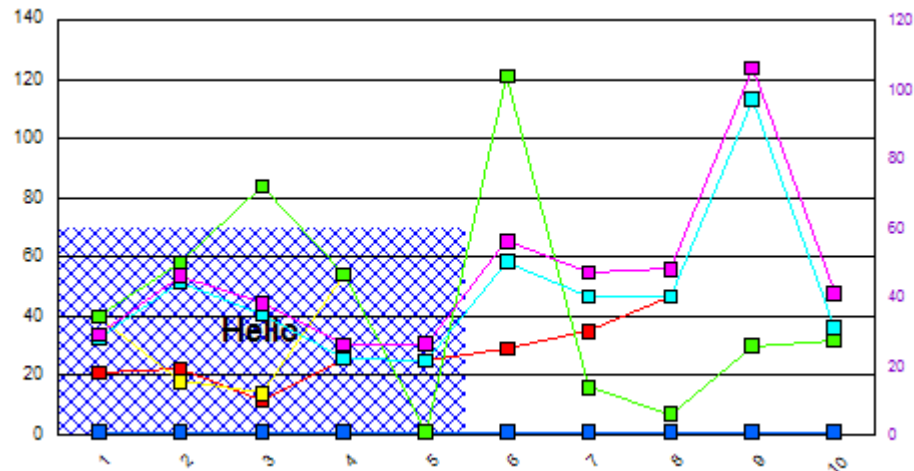
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



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

EXAMPLE:

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



PERSISTENT:

NO

NOTES:

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

@USER_FILL_CIRCLE (Color-Filled Circle)

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

SYNTAX:

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

PARAMETERS:

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

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

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

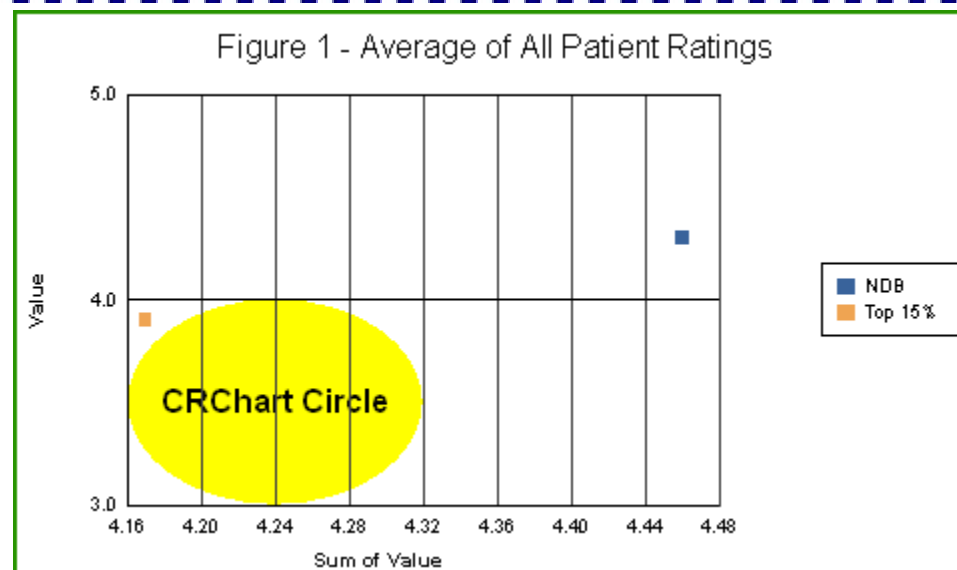
fStopY; 0.0...1.1 Y-Axis stop location

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

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

EXAMPLE:

```
@USER_FILL_CIRCLE 0.0 0.5 0.0 0.5 255 255 0 CRChart Circle
```

**PERSISTENT:**

NO

NOTES:

- A maximum of 5 @USER_CIRCLE, @USER_FILL..., and/or @USER_RECT macros may be specified but any combination of the macros may be used. @USER_CIRCLE, @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.
- This macro requires Crystal Reports 9.x or higher.

@USER_FILL_CIRCLE_ABOVE (Color-Filled Circle above Chart Area)

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

SYNTAX:

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

PARAMETERS:

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

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

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

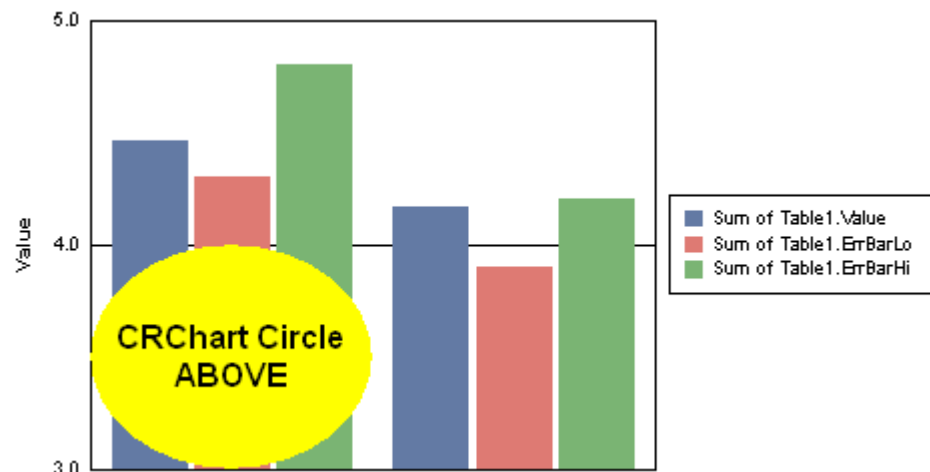
fStopY; 0.0...1.1 Y-Axis stop location

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

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

EXAMPLE:

```
@USER_FILL_CIRCLE_ABOVE 0.0 0.5 0.0 0.5 255 255 0 CRChart
Circle ABOVE
```



PERSISTENT:

NO

NOTES:

- A maximum of 5 @USER_CIRCLE, @USER_FILL..., and/or @USER_RECT macros may be specified but any combination of the macros may be used. @USER_CIRCLE, @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.
- This macro requires Crystal Reports 9.x or higher.

@USER_FILL_CIRCLE2 (Pattern-Filled Circle)

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

SYNTAX:

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

PARAMETERS:

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

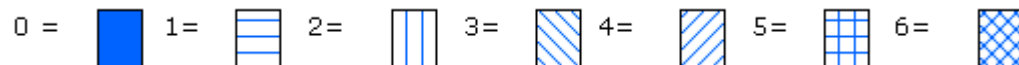
fStopX; 0.0...1.1 X-Axis stop location

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

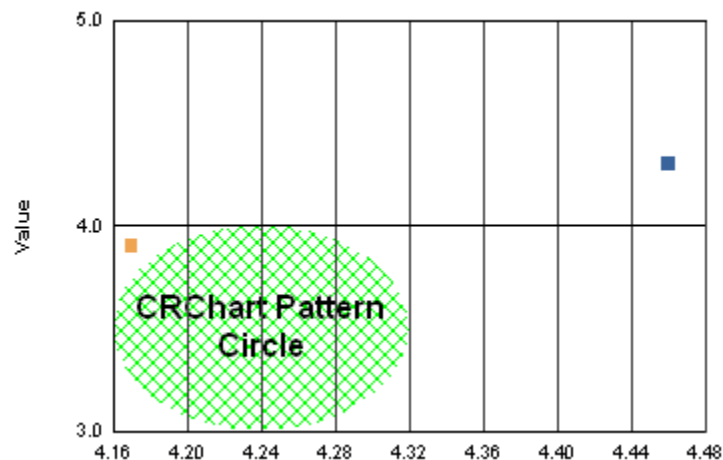
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



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

EXAMPLE:

```
@USER_FILL_CIRCLE2 0.0 0.5 0.0 0.5 0 255 0 6 CRChart Pattern
Circle
```

**PERSISTENT:**

NO

NOTES:

- A maximum of 5 @USER_CIRCLE, @USER_FILL..., and/or @USER_RECT macros may be specified but any combination of the macros may be used. @USER_CIRCLE, @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.
- This macro requires Crystal Reports 9.x or higher.

@USER_FILL_CIRCLE2_ABOVE (Pattern-Filled Circle above Chart Area)

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

SYNTAX:

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

PARAMETERS:

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

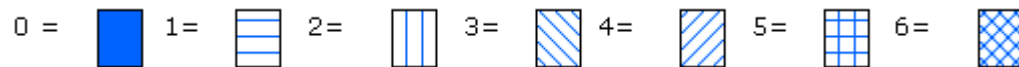
fStopX; 0.0...1.1 X-Axis stop location

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

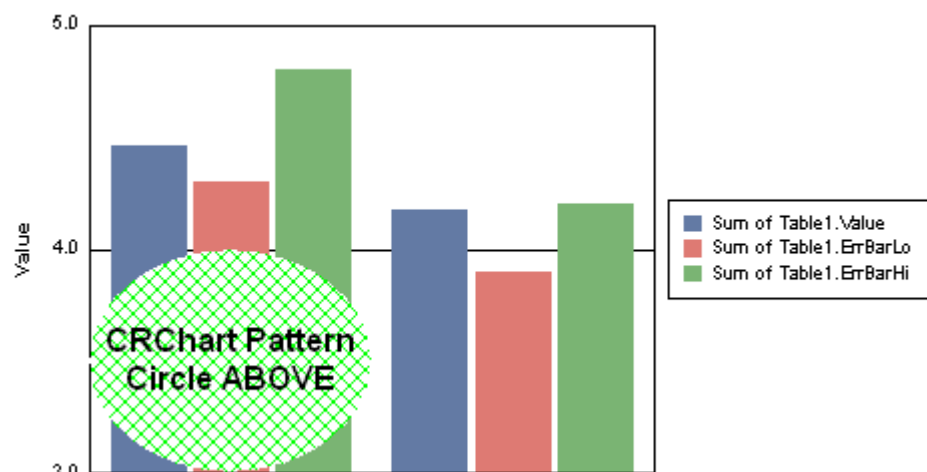
nPattern; -6...6. Positive value show one of the following patterns with a white background. Negative values show one of the following patterns with a transparent background.



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

EXAMPLE:

```
@USER_FILL_CIRCLE2_ABOVE 0.0 0.5 0.0 0.5 0 255 0 6 CRChart
Pattern Circle ABOVE
```



PERSISTENT:

NO

NOTES:

- A maximum of 5 @USER_CIRCLE, @USER_FILL..., and/or @USER_RECT macros may be specified but any combination of the macros may be used. @USER_CIRCLE, @USER_FILL..., and @USER_RECT are independent of the X or Y axis values/scales.
- This macro requires Crystal Reports 9.x or higher.

@USER_MARKER (User-Defined Marker)

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









SYNTAX:

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

PARAMETERS:

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

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

0=  1=  2=  3= 
 4=  5=  6=  7= 

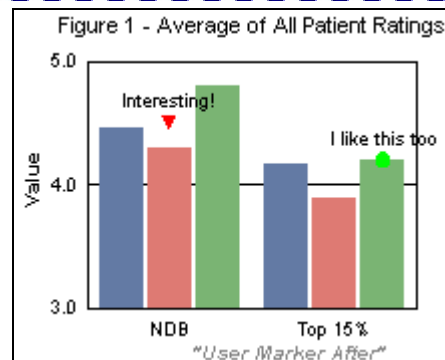
nRed; 0..255, *nGreen*; 0..255, *nBlue*; 0..255 define the color of the marker.

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

EXAMPLE:

This example draws a red (255 0 0) Triangle "tip-down" marker (7) at the X/Y coordinates 1.0/4.5 with the text string "Interesting!". A bright green (0 255 0) circle marker (1) is drawn at the X/Y coordinates 2.3/4.2 with the text string "I like this too".

```
@USER_MARKER 1.0 4.5 7 255 0 0 Interesting!  
@USER_MARKER 2.3 4.2 1 0 255 0 I like this too
```

**PERSISTENT:**

NO

NOTES:

Only 8 USER_MARKERS may be specified.

@USER_MARKER2 (User-Defined Marker with Value)

This macro adds a user-defined marker with optional text to a chart. With this macro, you can also specify a null marker and a value to append to the phrase (*szPhrase*). *fX* must be value that is between the minimum and maximum values shown on the X-Axis. *fY* must be value that is between the minimum and maximum values shown on the Y-Axis.









SYNTAX:

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

PARAMETERS:

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

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

0=  1=  2=  3= 
 4=  5=  6=  7= 

8=null/no marker

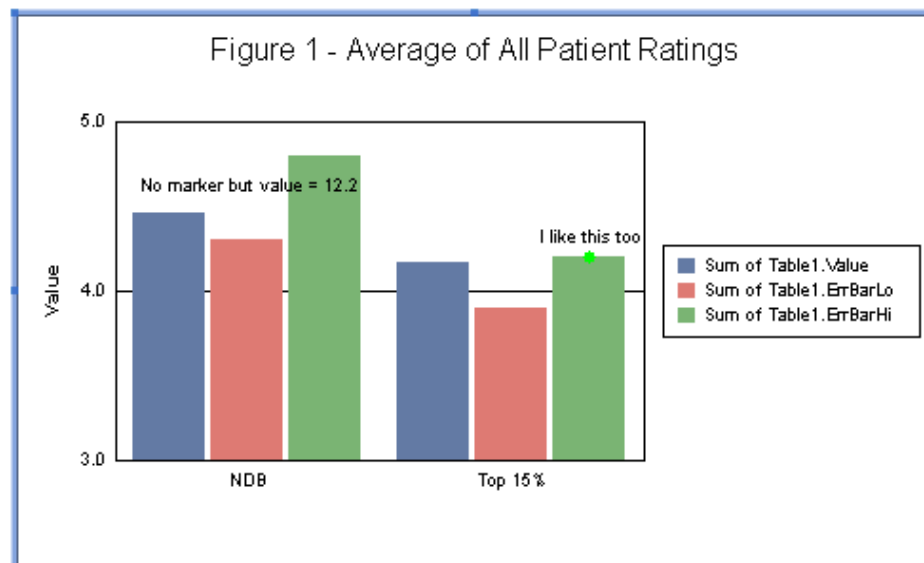
nRed; 0..255, *nGreen*; 0..255, *nBlue*; 0..255 define the color of the marker.

fValue; Value to append to *phrase*.

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

EXAMPLE:

```
@USER_MARKER2 1.0 4.5 8 255 0 0 12.2 No marker but value =  
@USER_MARKER 2.3 4.2 1 0 255 0 I like this too
```



PERSISTENT:

NO

NOTES:

- Only 8 USER_MARKERS may be specified.
- This macro requires Crystal Reports 9.x or higher.

@USER_RECT (Outlined Rectangle)

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

SYNTAX:

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

PARAMETERS:

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

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

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

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

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

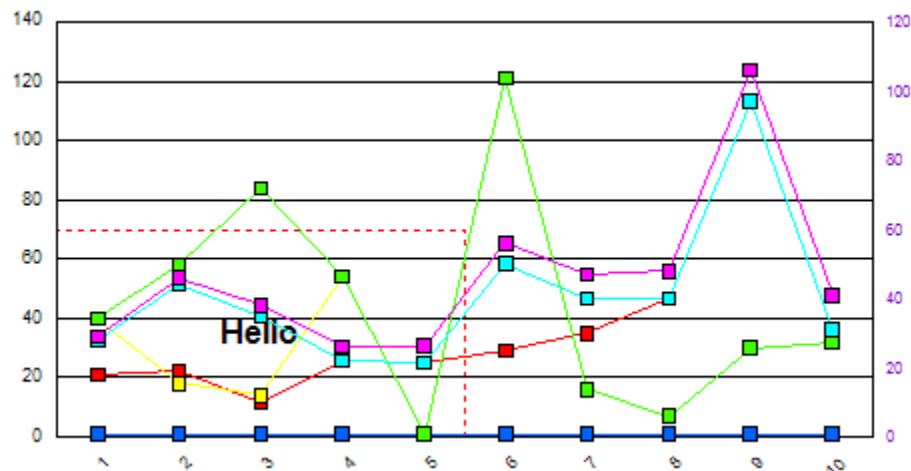
nLineStyle; 0...15 selects the line style (0=Solid, 1=Dashed, 2=Dotted, 3=Dot-Dash, 4=Dash-Dot-Dot, 5=Medium Dash, 6=Short Dash, 7=Long Dash, 8=Long Dot, 9=Dot-Dot-Dot, 10=Dash-Dash-Dot, 11=Dash-Dash-Dot-Dot, 12=Long Dash-Dot, 13=Long Dash-Dot-Dot, 14=Long Dash-Dash-Dot, 15=Long Dash-Dash-Dot-Dot)

nLineThickness; 0...1000 Line Thickness

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

EXAMPLE:

```
@USER_RECT 0.0 0.5 0.0 0.5 255 0 0 2 8 Hello
```



PERSISTENT:

NO

NOTES:

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

Section 11: Chart Type & Chart Specific Macros

These macros change the chart type:

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

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

ALSO SEE:

11.1 Macros for Box Plots

11.2 Macros for Gauges

11.3 Macros for Pie Charts

11.4 Macros for Waterfall Charts

@3DSCAT (3D Scatter Chart)

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

SYNTAX:

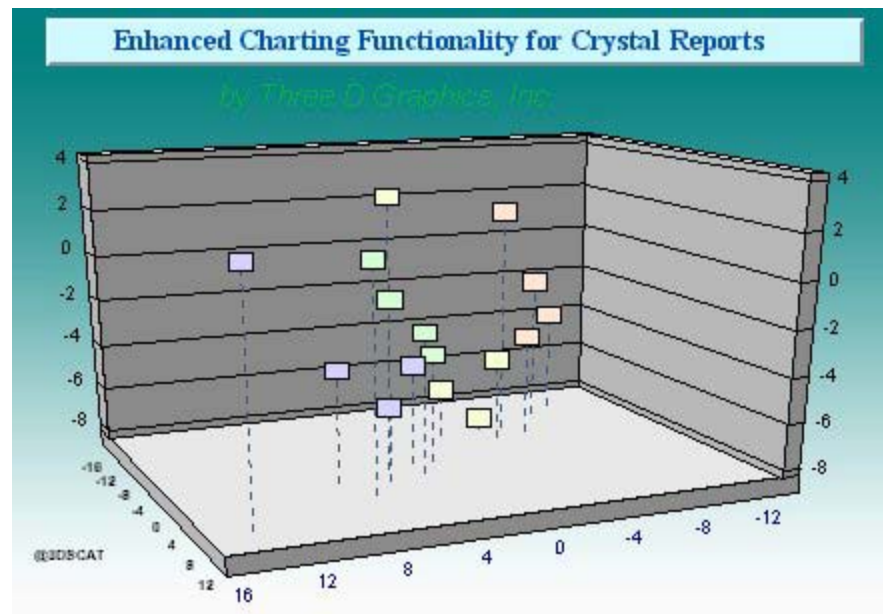
@3DSCAT

PARAMETERS:

None

EXAMPLE:

@3DSCAT



PERSISTENT:

YES

@AUDIO (Audiogram)

This macro creates an audiogram chart.

SYNTAX:

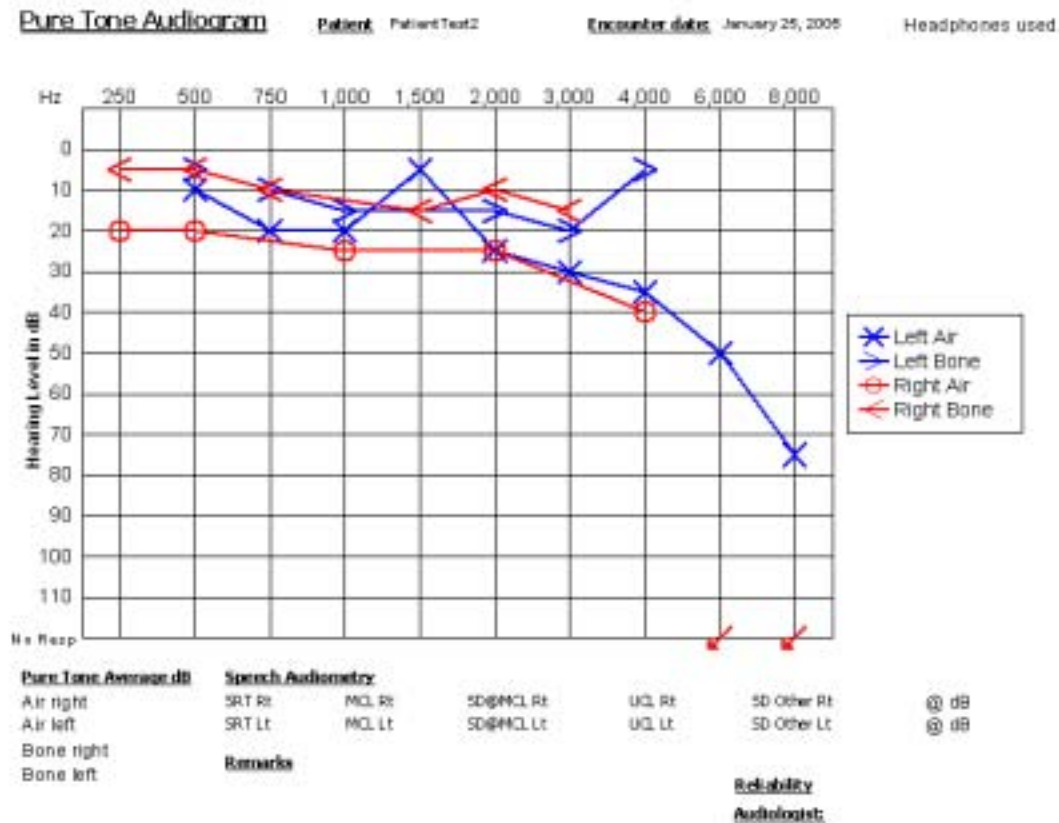
@AUDIO

PARAMETERS:

None

EXAMPLE:

@AUDIO

**PERSISTENT:**

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.

@COMBO (Combo Chart)

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

SYNTAX:

```
@COMBO nSeries nStyle
```

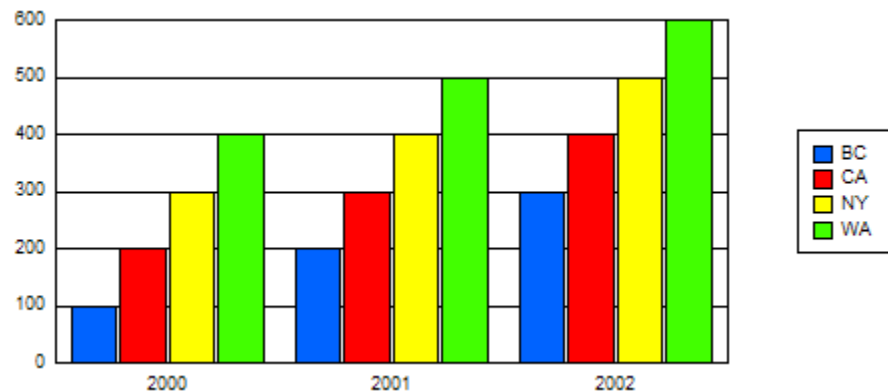
PARAMETERS:

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.

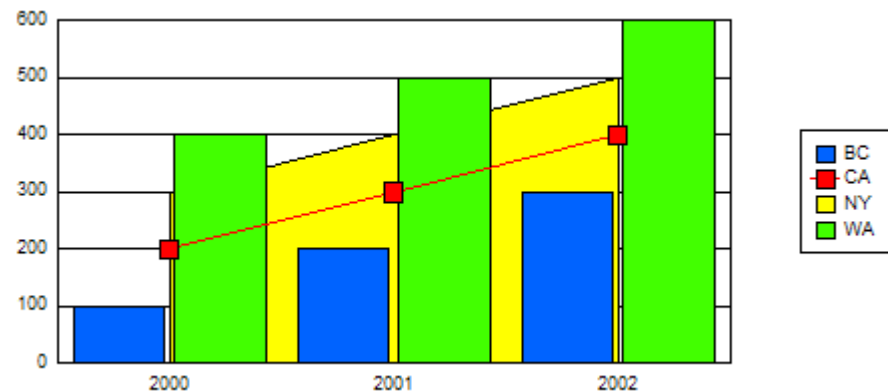
nStyle; 0..2 (0= Bar, 1=Line, 2=Area)

EXAMPLE:

```
@COMBO -1 0
```



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



PERSISTENT:

YES

@COMPARE2 (2-Series Absolute Bar Chart)

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

SYNTAX:

```
@COMPARE2 bCompare
```

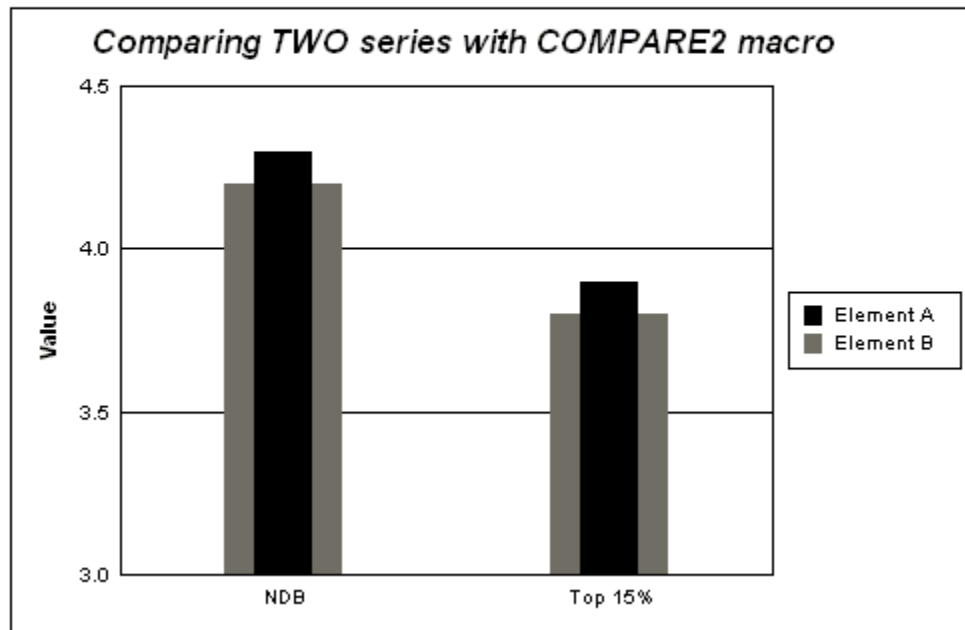
PARAMETERS:

bCompare; 0/1

EXAMPLE:

```
@COMPARE2 1
```

<u>ID</u>	<u>Segment</u>	<u>Value</u>	<u>ErrBarLo</u>	<u>ErrBarHi</u>
1	NDB	4.46	4.30	4.80
2	Top 15%	4.17	3.90	4.20

**PERSISTENT:**

YES

NOTES:

This macro requires Crystal Report 10.x or higher.

@GANTT (Gantt Chart)

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

SYNTAX:

```
@GANTT bTimeAxis bDurationMode bGroupCompress
```

PARAMETERS:

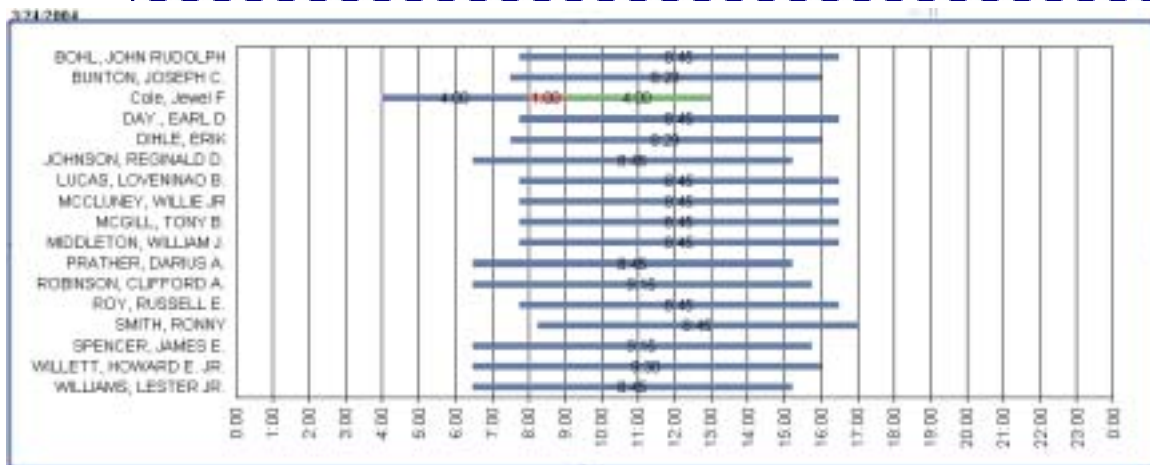
bTimeAxis; 1 = use time axis, 0 = do not use time axis

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

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

EXAMPLE:

```
@GANTT 1 1 1
```



PERSISTENT:

YES

@GRAPHTYPE (Graph Type)

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

SYNTAX:

@GRAPHTYPE *n*

PARAMETERS:

n; 0...90

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
31	Vertical Line Bi-Polar Stacked
32	Vertical Line Dual-Y Absolute
33	Vertical Line Dual-Y Stacked

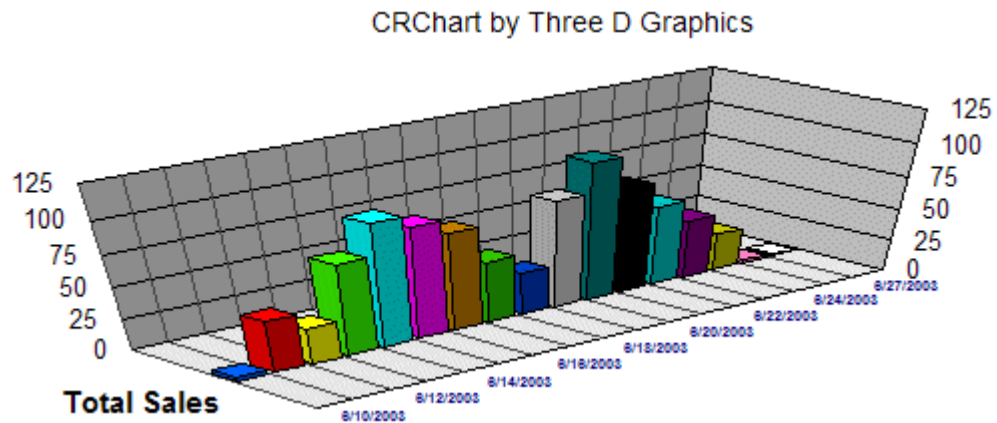
Chart Enhancement Macros for Crystal Reports

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

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

EXAMPLE:

@GRAPHTYPE 73



PERSISTENT:

YES

@PARETO (Pareto Chart)

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

SYNTAX:

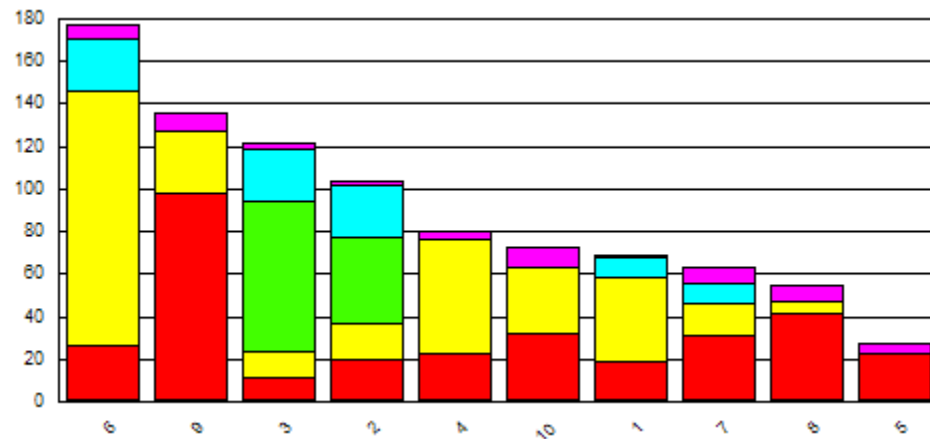
```
@PARETO n
```

PARAMETERS:

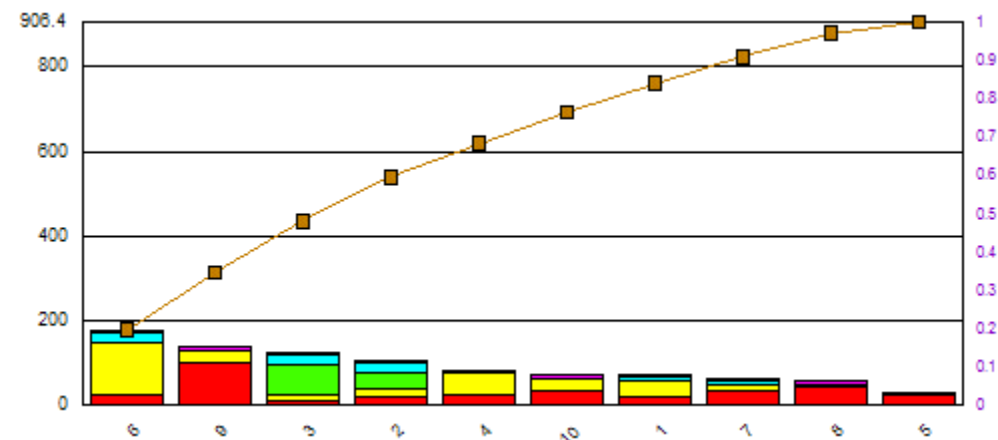
n ; 0 = create a simple pareto chart. 1 = create a "classic" pareto chart with a cumulative percentage line.

EXAMPLE:

```
@PARETO 0
```



```
@PARETO 1
```



PERSISTENT:

YES

@POLAR (Polar Chart)

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

SYNTAX:

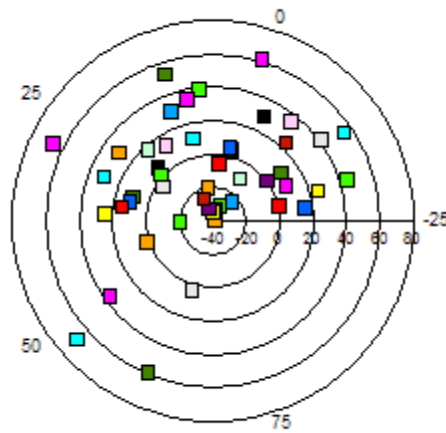
```
@POLAR
```

PARAMETERS:

None

EXAMPLE:

```
@POLAR
```

**PERSISTENT:**

YES

11.1 Macros for Box Plots

These macros create and modify box plot charts:

- @BP/BP1; Create a box plot chart with a square-style tail
- @BP2; Create a box plot chart with a T-style tail
- @BP3; Create a box plot chart with a I-style tail
- @BPH; Change the orientation of a Box Plot chart
- @BPW; Use the sixth data value to determine the width of Box Plots
- @IN; Move the First Box Plot In/Right by a specified number of virtual units
- @MC; Define Marker Colors in Box Plots
- @MK; Define the Number of Markers in Box Plots
- @MS; Define Marker Shapes in Box Plots
- @SZ; Defines the Size of Markers in Scatter or Box Plot charts

@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:

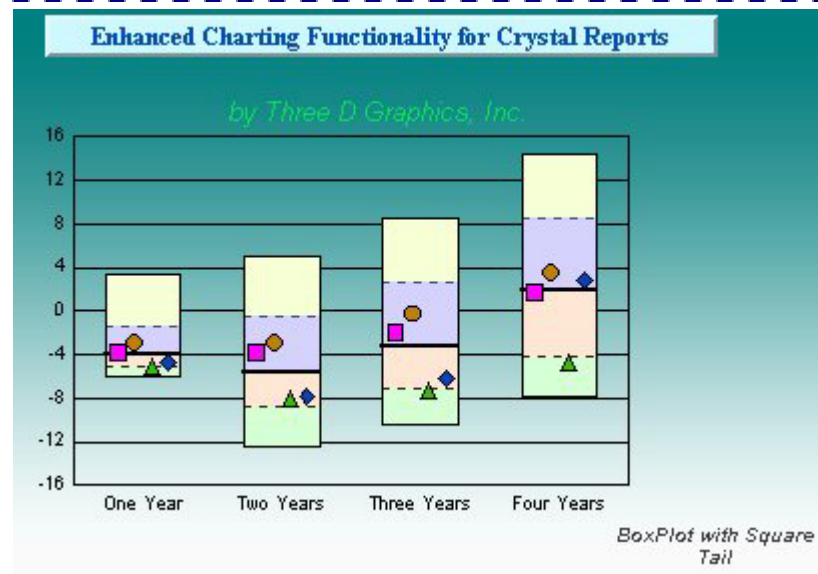
None

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	■	Series 2	●
Series 3	▲	Series 4	◆
Series 5	★	Series 6	✚
Series 7	✖	Series 8	▼

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:

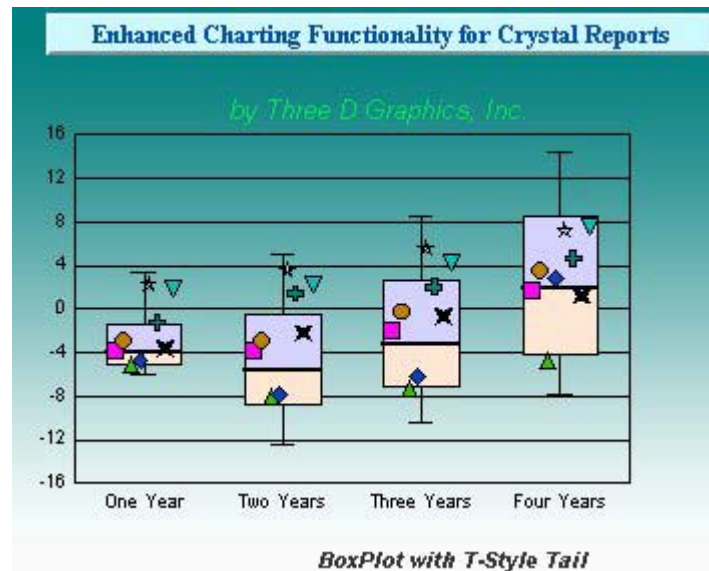
None

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	■	Series 2	●
Series 3	▲	Series 4	◆
Series 5	★	Series 6	+
Series 7	✕	Series 8	▼

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:

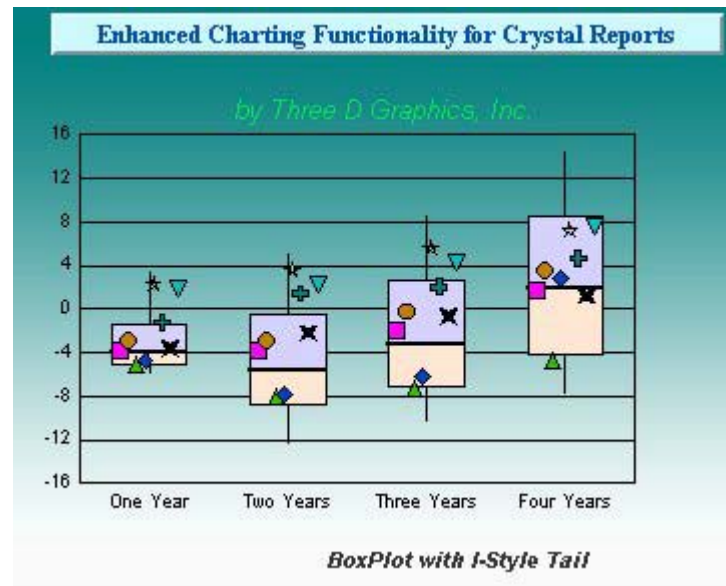
None

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	■	Series 2	●
Series 3	▲	Series 4	◆
Series 5	★	Series 6	+
Series 7	✕	Series 8	▼

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

@BPH (Horizontal Box Plot)

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

SYNTAX:

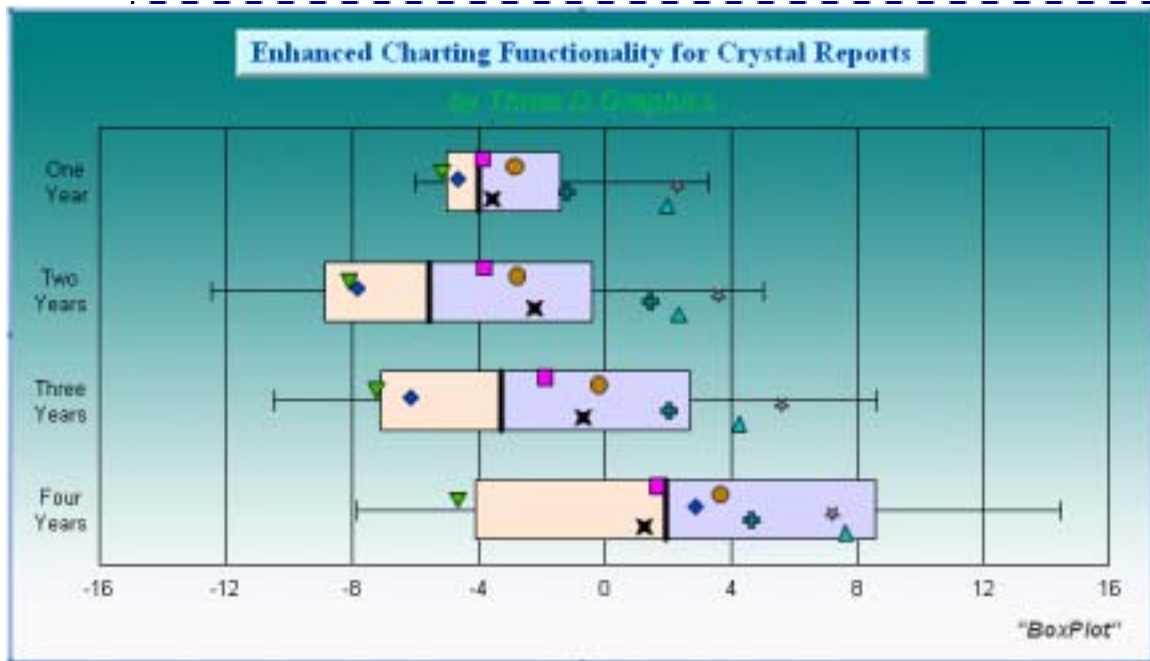
```
@BPH nOrient
```

PARAMETERS:

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

EXAMPLE:

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



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 10.x or higher.

@BPW (Box Plot Width)

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

SYNTAX:

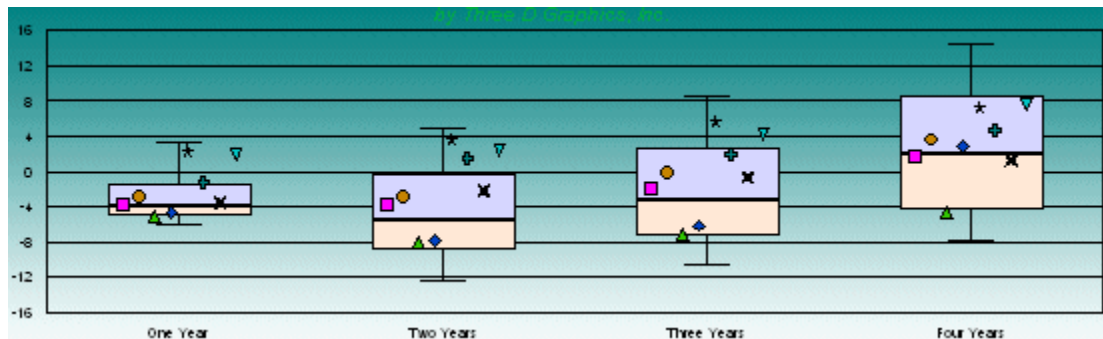
@BPW

PARAMETERS:

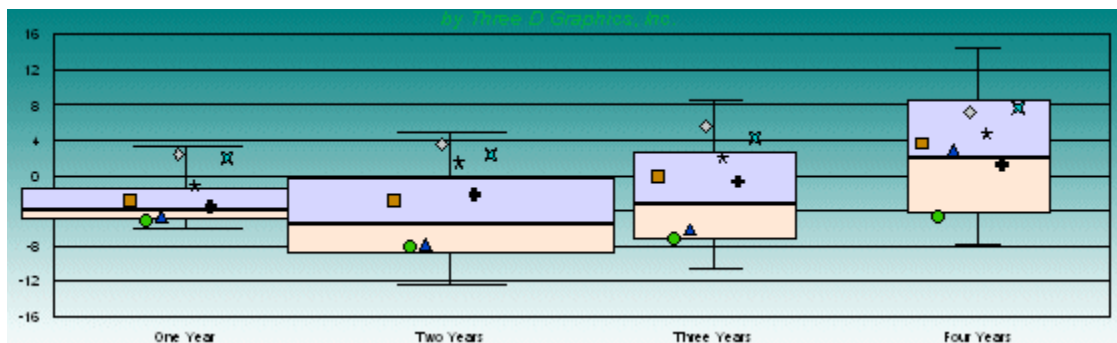
None

EXAMPLE:

Before @BPW



After @BPW



PERSISTENT:

NO

ALSO SEE:

@BP, @BP1, @BP2, @BP3

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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:

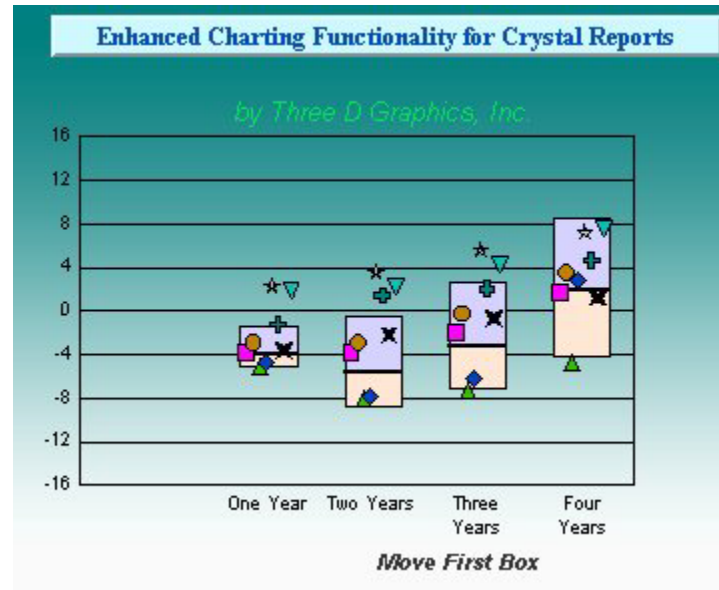
n; Number of virtual units (0...16000) to move the first box to the right.

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

@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)

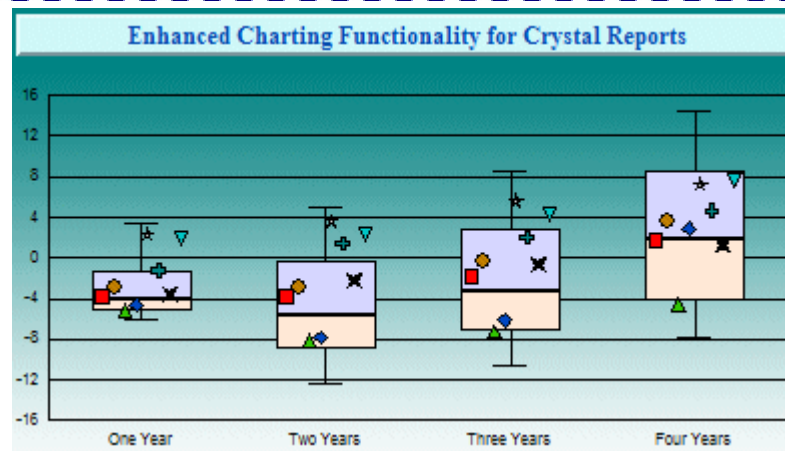
nR; 0...255 defines the Red portion of RGB color selection.

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 @GCOLOR macro to change the color of markers and risers in other chart types.

@MK (Number of Markers)

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

SYNTAX:

```
@MK n
```

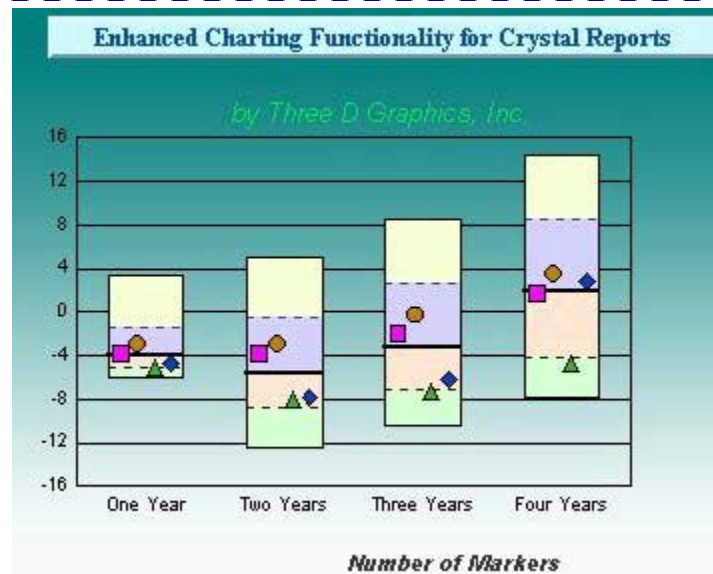
PARAMETERS:

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	■	Series 2	●
Series 3	▲	Series 4	◆
Series 5	★	Series 6	+
Series 7	✕	Series 8	▼

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.

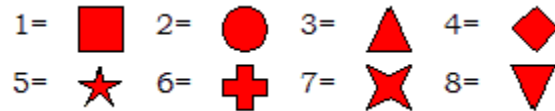
SYNTAX:

```
@MS n1 n2
```

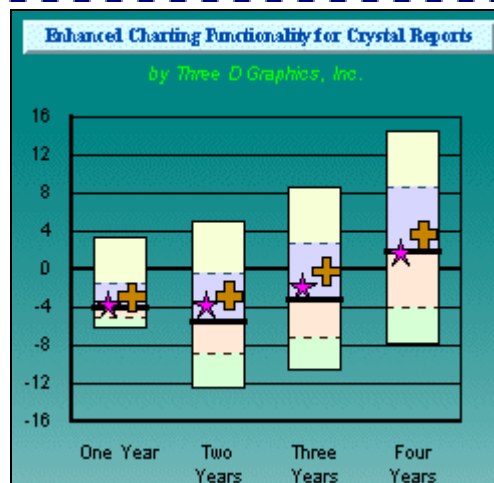
PARAMETERS:

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.

@SZ (Size of Markers)

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

SYNTAX:

```
@SZ n1
```

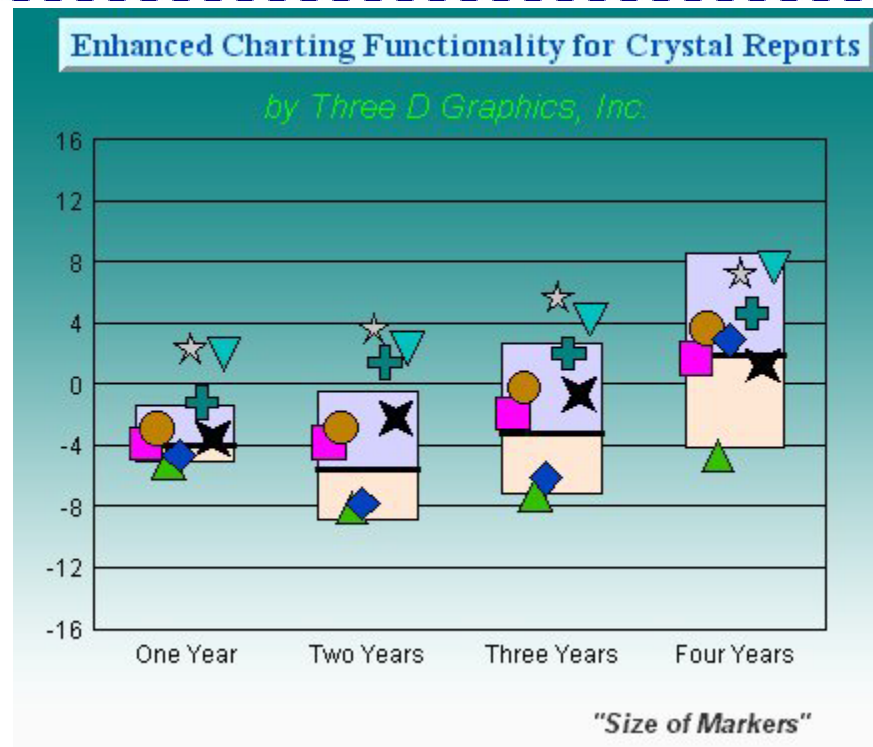
PARAMETERS:

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

11.2 Macros for Gauges

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

- @GAUGE_BORDER_STYLE; Select gauge border (**CHCHART PLUS**)
- @GAUGE_BORDER_THICKNESS; Control the thickness of the new gauge engine (**CHCHART PLUS**)
- @GAUGE_COLOR; Control the color of any band on the gauge (**CHCHART PLUS**)
- @GAUGE_MULTIPLE_NEEDLES; Draw all or only first series as a gauge needle (**CHCHART PLUS**)
- @GAUGE_NEEDLE_STYLE; Select a needle style (**CHCHART PLUS**)
- @GAUGE_STYLE; Enable/Disable new gauge drawing engine (**CHCHART PLUS**)
- @GAUGE_THRESHOLD; Define Gauge Band Thresholds

@GAUGE_BORDER_STYLE (*Gauge Border Style*)



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

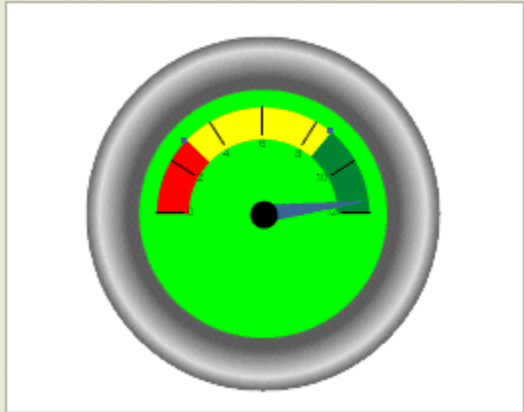
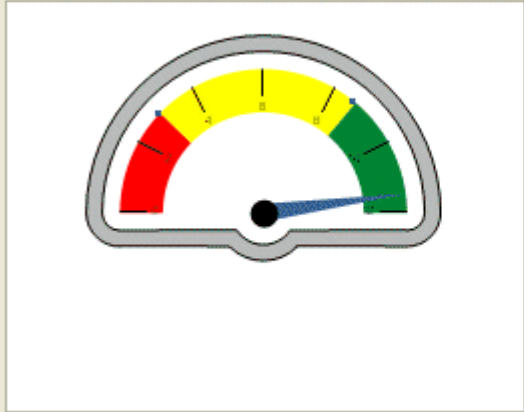
SYNTAX:

```
@GAUGE_BORDER_STYLE nStyle
```

PARAMETERS:

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

EXAMPLE:

	Title: <input type="text" value="@GAUGE_STYLE 1"/>
	Subtitle: <input type="text" value="@GAUGE_BORDER_STYLE 4"/>
	Footnote: <input type="text"/>
	Title: <input type="text" value="@GAUGE_STYLE 1"/>
	Subtitle: <input type="text" value="@GAUGE_BORDER_STYLE 6"/>
	Footnote: <input type="text"/>

PRESISTENT:

NO

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GAUGE_BORDER_THICKNESS (*Gauge Border Thickness*)



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

SYNTAX:

@GAUGE_BORDER_THICKNESS *nThick*

PARAMETERS:

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

EXAMPLE:

	<p>Title: <input type="text" value="@GAUGE_STYLE 1"/></p> <p>Subtitle: <input type="text" value="@GAUGE_BORDER_THICKNESS 100"/></p> <p>Footnote: <input type="text" value=""/></p>
	<p>Title: <input type="text" value="@GAUGE_STYLE 1"/></p> <p>Subtitle: <input type="text" value="@GAUGE_BORDER_THICKNESS 1"/></p> <p>Footnote: <input type="text" value=""/></p>

PRESISTENT:

NO

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GAUGE_COLOR (Color Gauge Bands)



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

SYNTAX:

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

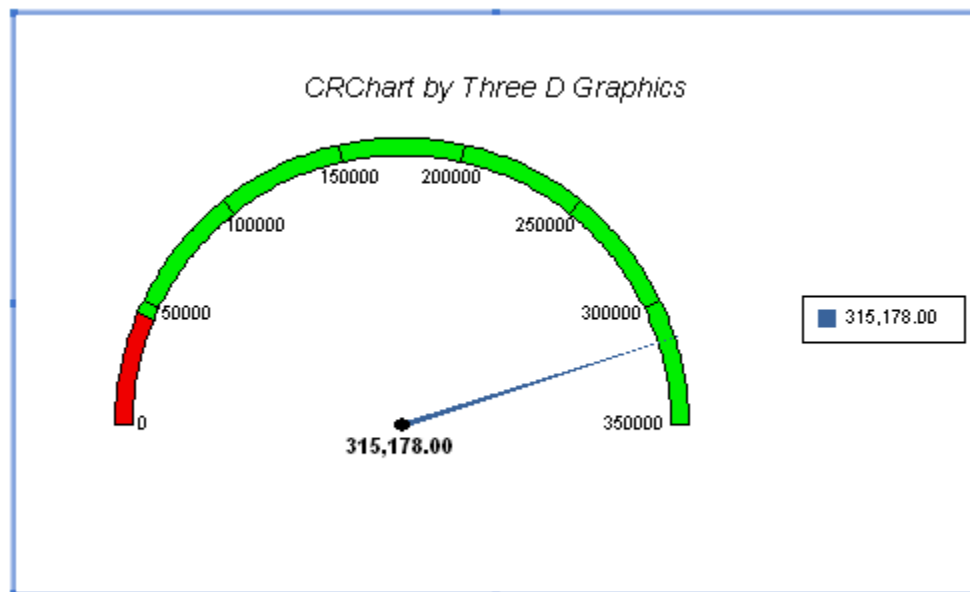
PARAMETERS:

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

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

EXAMPLE:

```
@GAUGE_COLOR 0 233 2 2  
@GAUGE_COLOR 1 2 232 2
```



PERSISTENT:

NO

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GAUGE_MULTIPLE_NEEDLES (Multiple Needles)

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

SYNTAX:

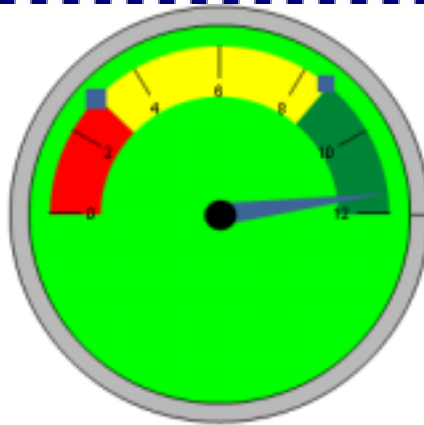
```
@GAUGE_MULTIPLE_NEEDLES bEnable
```

PARAMETERS:

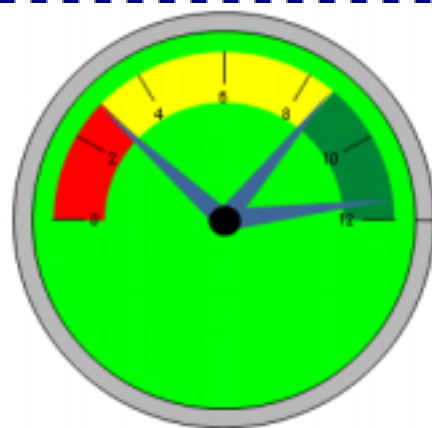
bEnable; 0=Draw first series as a needle, draw all other series as markers. 1=Draw all series as needles.

EXAMPLE:

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



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

**PRESISTENT:**

NO

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GAUGE_NEEDLE_STYLE (*Gauge Needle Style*)

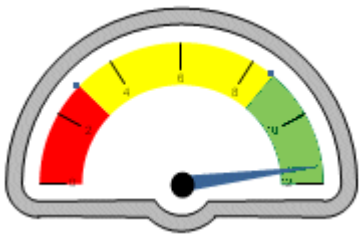
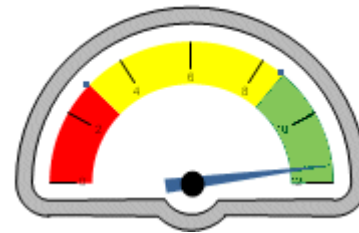
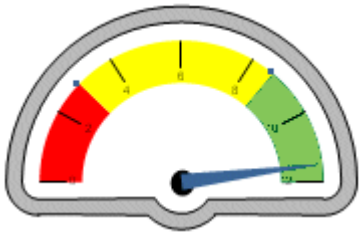
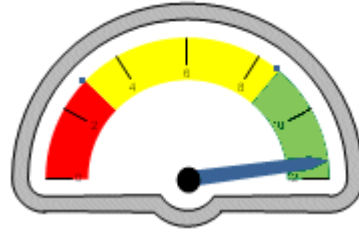
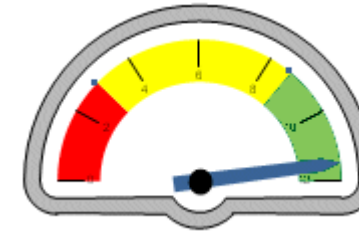
When gauge style is enabled (@GAUGE_STYLE 1), this macro can be used to select a needle style.

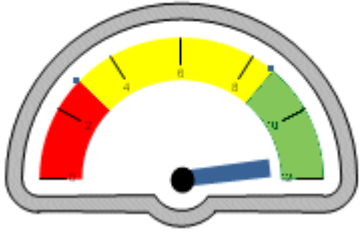
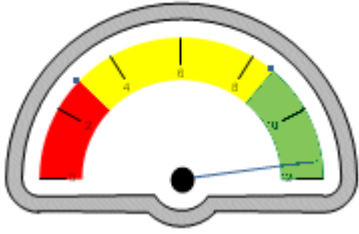
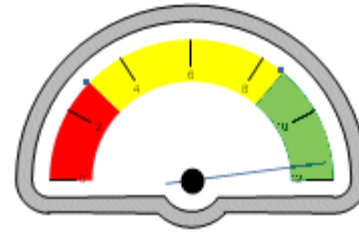
SYNTAX:

```
@GAUGE_NEEDLE_STYLE nStyle
```

PARAMETERS:

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

<i>nStyle</i>	Example	Description
0		Triangle
1		Triangle with extended end
2		Triangle above base
3		Steeple
4		Steeple with extended end

<i>nStyle</i>	Example	Description
5		Rectangle
6		Thin
7		Thin with extended end

PERSISTENT:

NO

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GAUGE_STYLE (Gauge Style)



This macro enables/disables the enhanced gauge drawing engine.

SYNTAX:

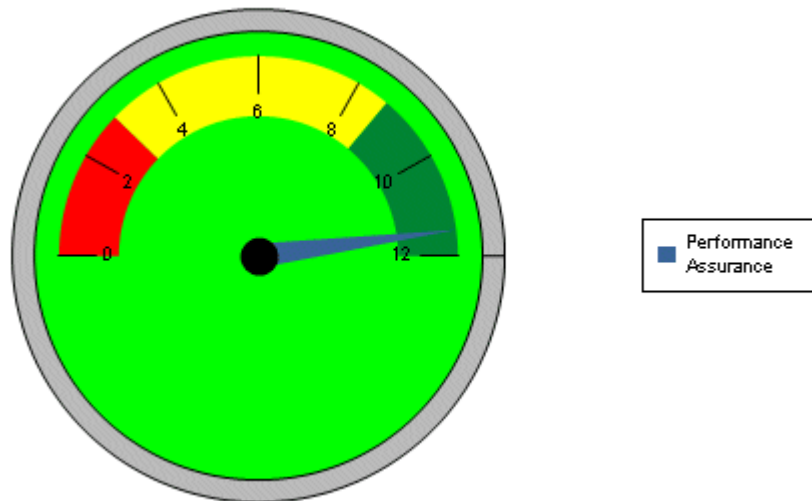
```
@GAUGE_STYLE bEnable
```

PARAMETERS:

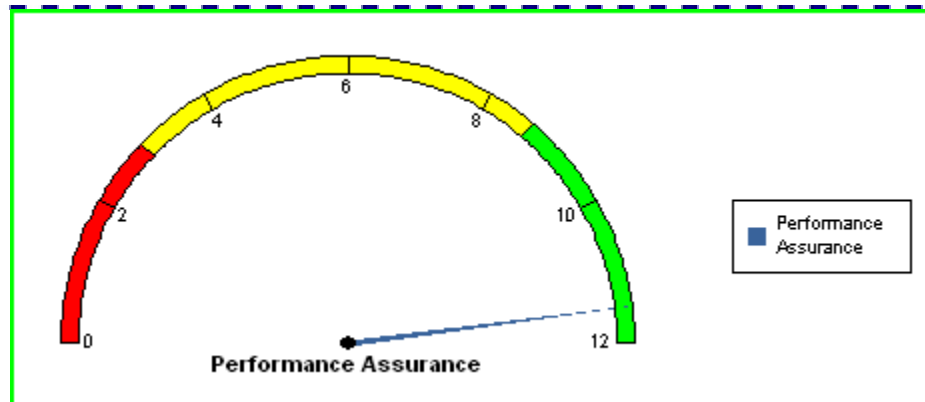
bEnable; 0=Use standard gauge drawing engine. 1=Use Enhanced Gauge Drawing Engine.

EXAMPLE:

```
@GAUGE_STYLE 1
```



```
@GAUGE_STYLE 0
```



PRESISTENT:

NO

NOTES:

- This macro requires CRChart Plus.
- This macro will automatically enable the high-quality drawing engine (@HQ 1).

@GAUGE_THRESHOLD (Gauge Band Threshold)

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

SYNTAX:

```
@GAUGE_THRESHOLD n1 f1
```

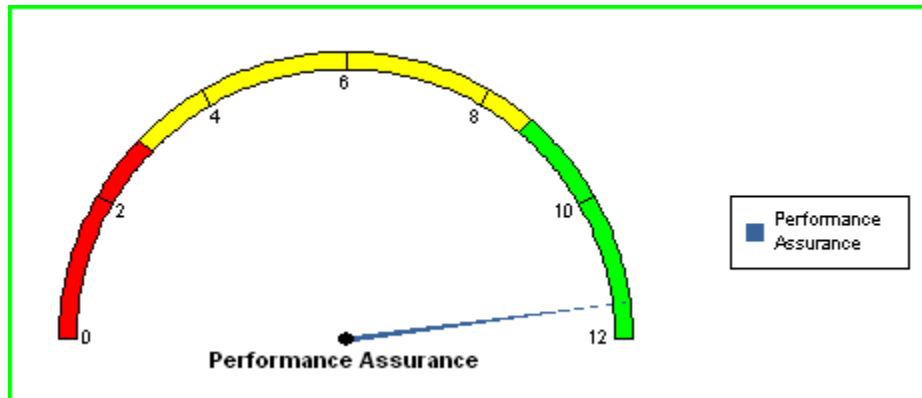
PARAMETERS:

n1; 0...4 selects the color band

f1; selects the threshold between color band *n1* and *n1*+1

EXAMPLE:

```
@GAUGE_THRESHOLD 0 P1
@GAUGE_THRESHOLD 1 P2
```

**PERSISTENT:**

NO

NOTES:

- This example is using Crystal Reports fields (P1, P2) to populate the *f1* parameter. See "Using Crystal Reports Fields/Functions in CRChart Macros" for more information about this functionality.
- This macro requires Crystal Reports 10.x or higher.

11.3 Macros for Pie Charts

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

- @DATATEXT_PIE; Control the appearance of data text in a pie chart
- @PIE_NEG; Enable handling of negative values in the legend area of a pie chart
- @SMART_PIE_LABELS; Enable/Disable pie label layout engine
(**CHCHART PLUS**)

@DATATEXT_PIE (Data Text Mode for Pie Charts)

This macro sets the data text mode in pie charts.

SYNTAX:

```
@DATATEXT_PIE n
```

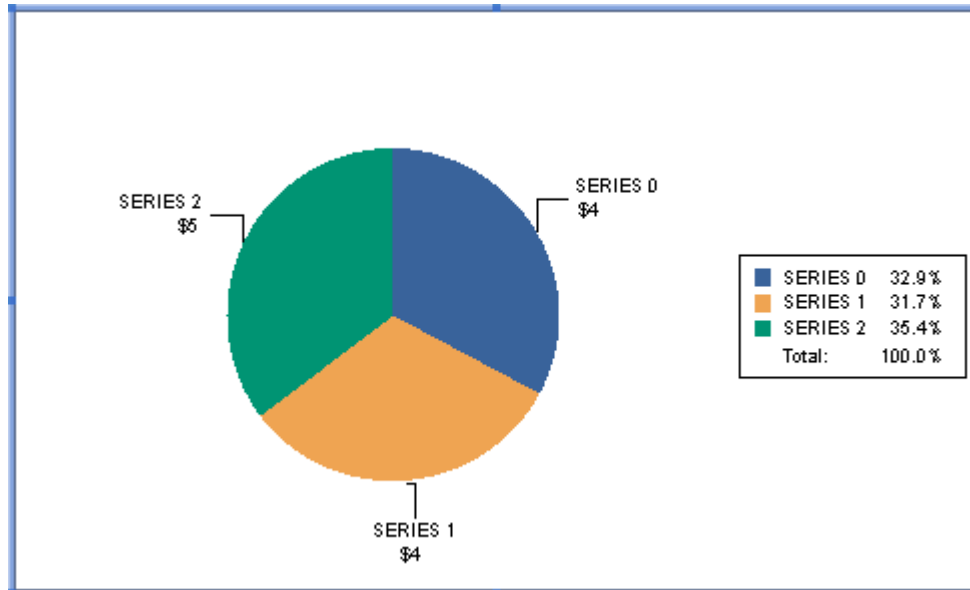
PARAMETERS:

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

- 0 = NO Labels on Pie
- 1 = VALUE labels on Pie
- 2 = Series Labels on Pie
- 3 = Value and Series Labels on Pie

EXAMPLE:

```
@DATATEXT_PIE 3
```

**PERSISTENT:**

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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

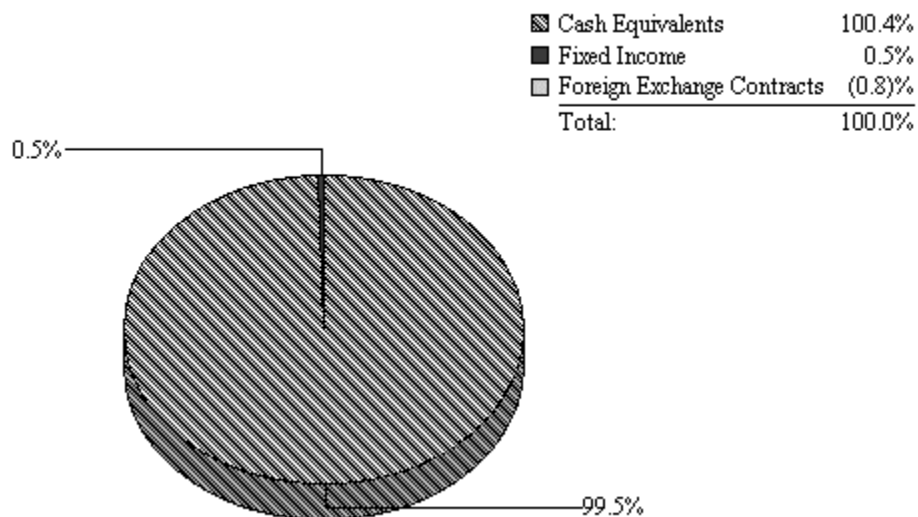
PARAMETERS:

None

EXAMPLE:

@PIE_NEG

Asset Allocation



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

@SMART_PIE_LABELS (Smart Pie Labels)

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

SYNTAX:

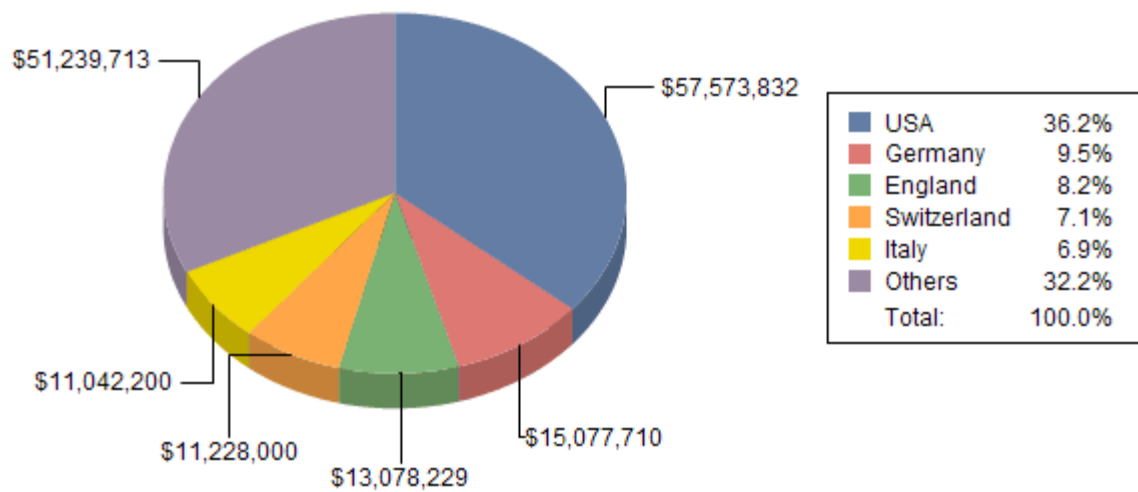
```
@SMART_PIE_LABELS bEnable
```

PARAMETERS:

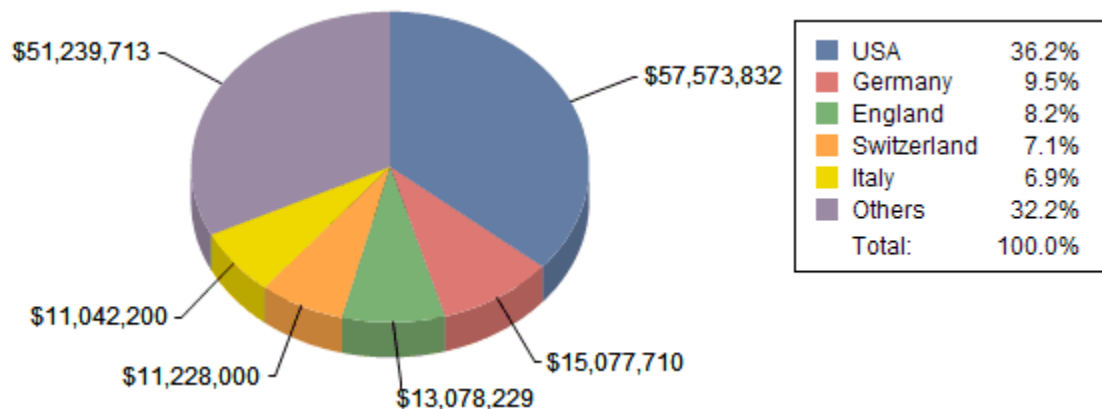
bEnable; 0=use standard pie engine, 1=use enhanced pie label layout engine

EXAMPLE:

```
@SMART_PIE_LABELS 0
```



```
@SMART_PIE_LABELS 1
```

**PRESISTENT:**

NO

NOTES:

This macro requires CRChart Plus.

11.4 Macros for Waterfall Charts

These macros create and format waterfall charts:

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

@WATERFALL (Waterfall Chart)

This macro creates a waterfall chart.

SYNTAX:

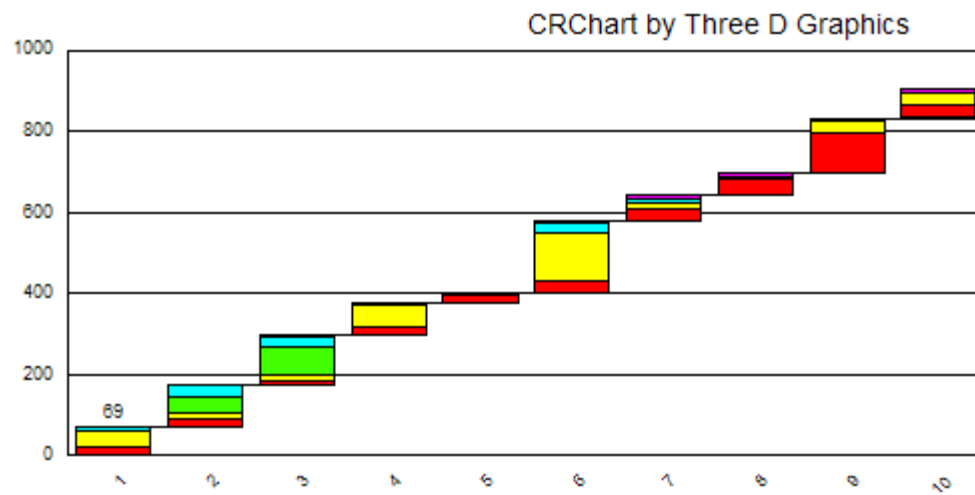
@WATERFALL

PARAMETERS:

None

EXAMPLE:

@WATERFALL

**PERSISTENT:**

YES

ALSO SEE:

@WATERFALL2

@WATERFALL2 (*Waterfall Chart with Total Group*)

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

SYNTAX:

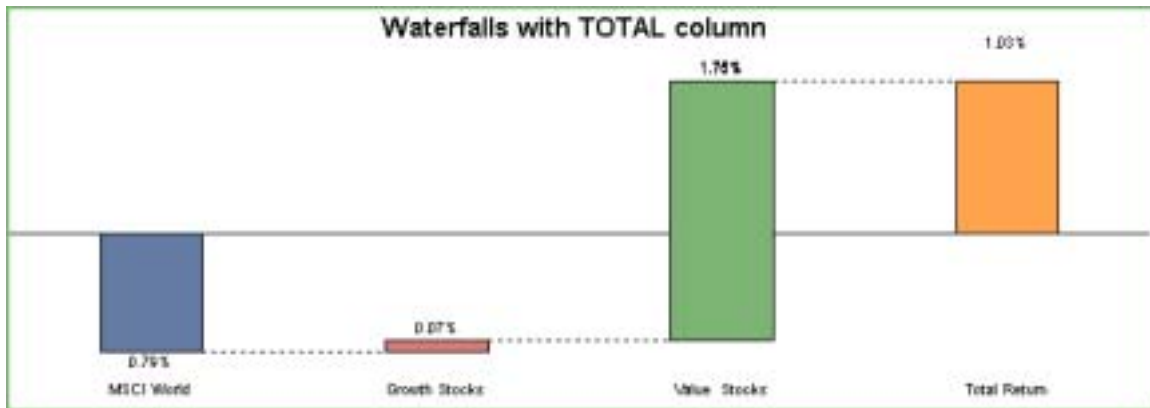
@WATERFALL2

PARAMETERS:

None

EXAMPLE:

@WATERFALL2



PERSISTENT:

YES

ALSO SEE:

@WATERFALL

NOTES:

This macro requires Crystal Reports 9.x or higher.

@WF_CONNECT (Waterfall Feeler Lines)

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

SYNTAX:

```
@WF_CONNECT nStyle
```

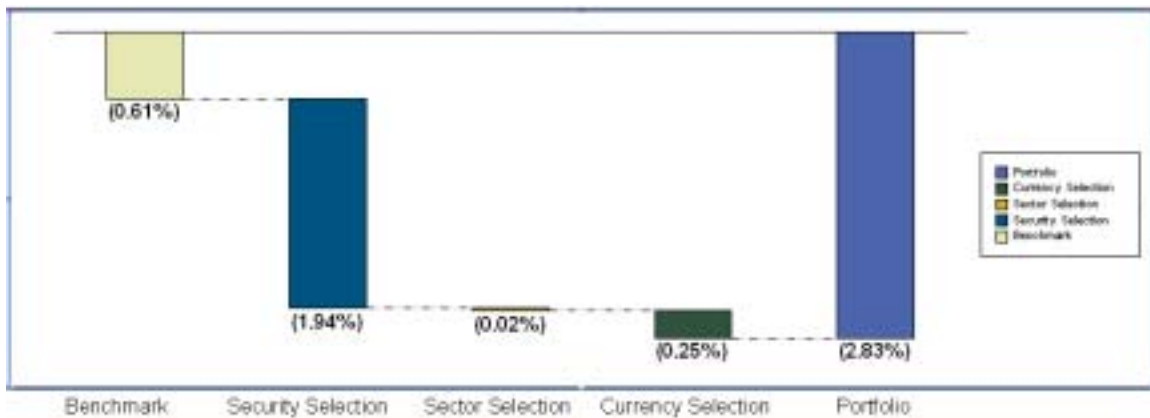
PARAMETERS:

nStyle; 0...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

EXAMPLE:

```
@WATERFALL2
@WF_CONNECT 4
@MCOLOR 4 74 98 168
@NEG_STYLE 1
```

**PERSISTENT:**

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.



Section 12: Miscellaneous

- @DEBUG; Show Debug Information
- @EB; Adds error bars to standard bar/column charts
- @FONTANGLE; Change the rotation angle of a text object
- @FONTSIZE; Change the font size of a text object
- @HAT; Define the size of hats on Error Bars
- @LEGEND_ORDER; Force Legend Order

@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

@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:

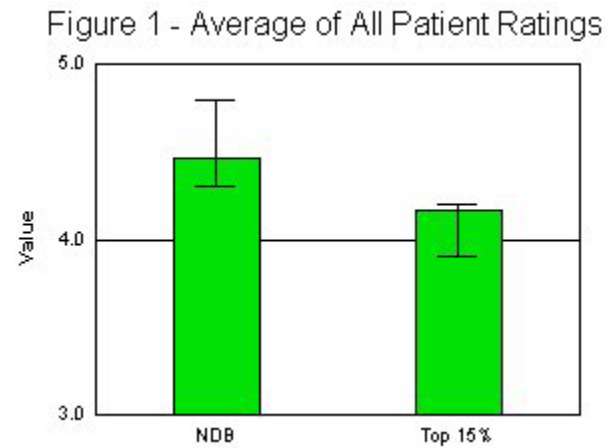
```
@EB n
```

PARAMETERS:

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

EXAMPLE:

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



PERSISTENT:

NO

ALSO SEE:

@HAT (Hat on Error Bars)

@FONTANGLE (Font Angle)

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

SYNTAX:

```
@FONTANGLE nObject nAngle
```

PARAMETERS:

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

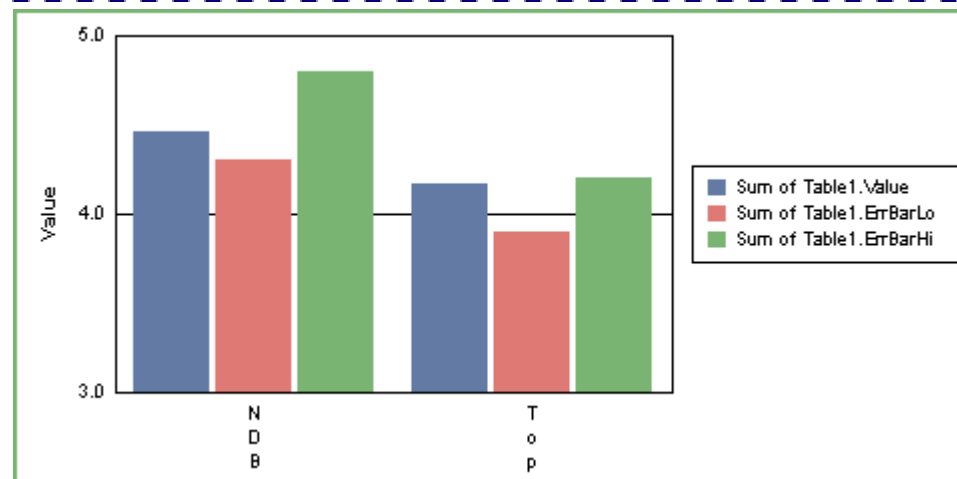
- 1 = Title
- 2 = Subtitle
- 3 = X1-Axis Title
- 4 = Y1-Axis Title
- 5 = Y2-Axis Title
- 6 = Footnote
- 7 = Data Text
- 8 = O1-Axis Labels or X1-Axis Labels depending on the chart type
- 9 = Y1-Axis Labels
- 10 = Y2-Axis Labels
- 11 = Legend Text

nAngle; 0...6 applies one of the following angles to *nObject*:

- 0 = Normal, Horizontal Characters
- 1 = Vertical/Hotel-Mode Characters
- 2 = Rotate Characters 90 Degrees
- 3 = Rotate Characters 180 Degrees
- 4 = Rotate Characters 270 Degrees
- 5 = Rotate Characters 45 Degrees
- 6 = Rotate Characters 315 Degrees

EXAMPLE:

```
@FONTANGLE 8 1
```



PERSISTENT:

YES

NOTES:

This macro requires Crystal Reports 9.x or higher.

@FONTSIZE (Font Size)

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

SYNTAX:

```
@FONTSIZE nObject nSize
```

PARAMETERS:

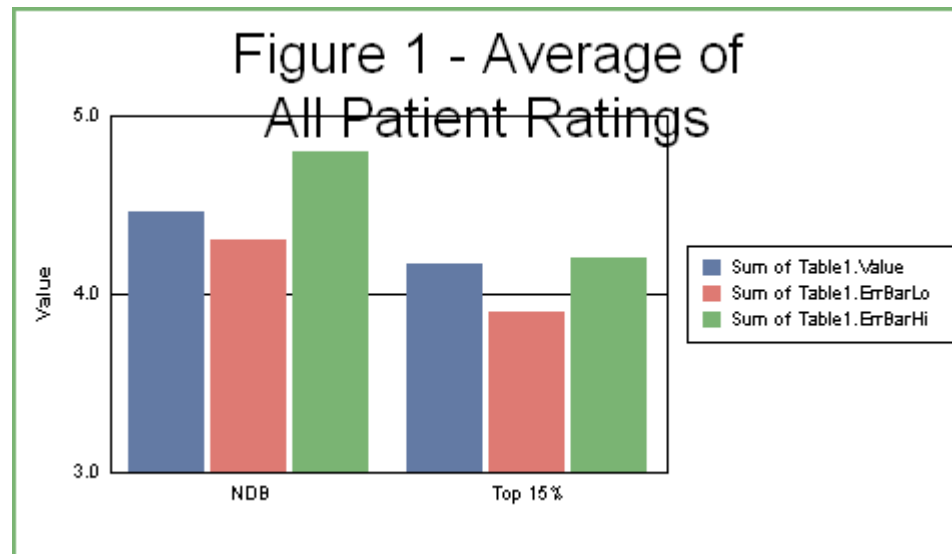
nObject; 1...11 selects one of the following chart objects:

- 1 = Title
- 2 = Subtitle
- 3 = X1-Axis Title
- 4 = Y1-Axis Title
- 5 = Y2-Axis Title
- 6 = Footnote
- 7 = Data Text
- 8 = O1-Axis Labels or X1-Axis Labels depending on the chart type
- 9 = Y1-Axis Labels
- 10 = Y2-Axis Labels
- 11 = Legend Text
- 12 = Series labels in Pie Charts
- 13 = Value labels in Pie Charts

nSize; any INT16 value that defines a font point size

EXAMPLE:

```
@FONTSIZE 1 20
```



PERSISTENT:

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

@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
```

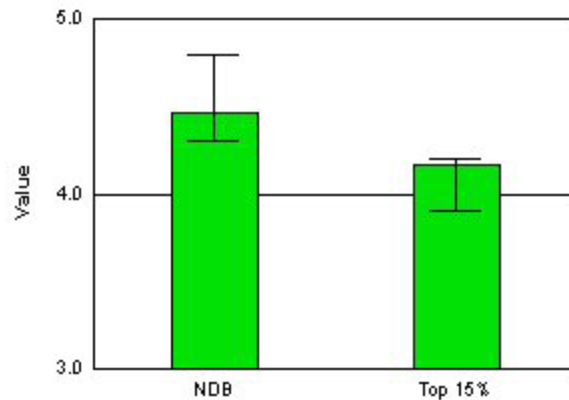
PARAMETERS:

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

EXAMPLE:

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

Figure 1 - Average of All Patient Ratings



"Hat on Error Bars"

PERSISTENT:

NO

ALSO SEE:

@EB

@LEGEND_ORDER (Force Legend Order)

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

SYNTAX:

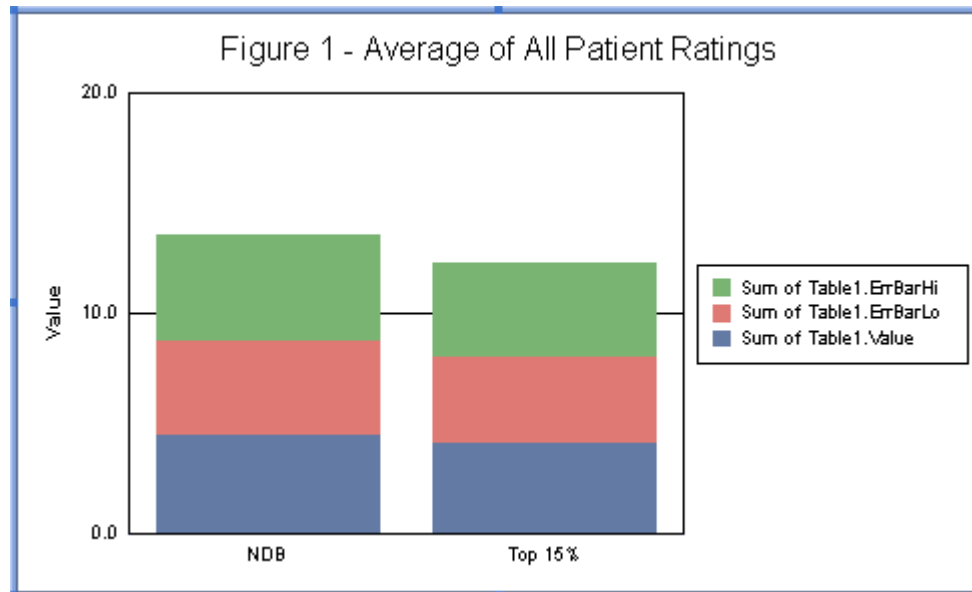
```
@LEGEND_ORDER nForceLegendOrder
```

PARAMETERS:

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

EXAMPLE:

```
@GRAPHTYPE 15  
@LEGEND_ORDER 2
```



PERSISTENT

NO

NOTES:

This macro requires Crystal Reports 9.x or higher.

