SAS® 9.4 Graph Template Language: Reference, Fifth Edition
# Contents

*What’s New in SAS 9.4 Graph Template Language* ........................................... vii

**PART 1  Fundamentals  1**

*Chapter 1  /  Overview* ........................................................................................................... 3
  *Graph Template Language (GTL)* .......................................................... 3
  *Components of a Graph* ................................................................................. 7
  *Graphical Layouts* .................................................................................. 8
  *Plots* ........................................................................................................... 10
  *Axes* ........................................................................................................... 11
  *Legends* .................................................................................................... 12
  *Flexible Templates* .................................................................................... 13
  *Output* ....................................................................................................... 16
  *About the Examples in This Documentation* ........................................... 18
  *Examples and Resources on the Web* ................................................... 19

**PART 2  Graph Block  21**

*Chapter 2  /  BEGINGRAPH Statement* ............................................................................. 23
  *Dictionary* .................................................................................................. 23

**PART 3  Layout Statements  41**

*Chapter 3  /  Summary of Layout Statements* ........................................................................ 43
  *Overview* ................................................................................................... 43
  *Single-cell Layouts* ....................................................................................... 44
  *Multi-cell Layouts* ...................................................................................... 45
  *Data-driven Layouts* .................................................................................... 46
  *Legend Layout* ........................................................................................... 46

*Chapter 4  /  Layout Statements* ......................................................................................... 47
  *Dictionary* .................................................................................................. 47
### PART 4  Plot Statements  187

**Chapter 5 / Key Concepts for Using Plots**
- Minimum Requirements to Generate a Plot  189
- ODS Graphics Environment  190
- Display Attributes  191
- Location and Position of Curve Labels  200

**Chapter 6 / Plot Statements**
- Dictionary  204

### PART 5  Plot Axes  971

**Chapter 7 / Axis Features in Layouts**
- Overview  973
- How Axis Features Are Determined  973
- Controlling Axis Features  982

**Chapter 8 / Axis Options in Layouts**
- Dictionary  987

### PART 6  Legend Statements  1211

**Chapter 9 / Legend Statements**
- Dictionary  1213

### PART 7  Text Statements  1263

**Chapter 10 / Managing Text Items**
- Overview  1265
- Using Prefix Options  1266
- Using Text Commands  1269
- Reserved Keywords and Unicode Values  1271

**Chapter 11 / Text Statements**
- Dictionary  1275

### PART 8  Custom Marker Definition Statements  1301

**Chapter 12 / Custom Marker Definition Statements**
- Dictionary  1303
What’s New in SAS 9.4 Graph Template Language

Overview

New and enhanced statements for the Graph Template Language (GTL) extend the versatility of the language and introduce new plot types. The following changes are included in this release:

- new plot statements
- new features for general use
- enhancements to SAS 9.3 statements

Possible Changes Required for Your Existing SAS GTL Programs

There are changes in GTL for SAS 9.4 and subsequent maintenance releases that might require you to modify your existing SAS GTL programs. Review this section carefully to determine whether you need to modify your existing SAS GTL programs for SAS 9.4.

Programs That Use the BARCHART or BARCHARTPARM Statement

In SAS 9.4, the BARCHART statement STAT=PCT option now displays percentages in the range 0–100 in order to be consistent with other GTL statements. In prior releases, STAT=PCT displays proportional values in the range 0–1. To restore the proportional values in SAS 9.4 and later releases, change STAT=PCT to STAT=PROPORTION.

SAS 9.4M3 adds option COLORSTAT= to the BARCHART statement. It is enabled by the COLORRESPONSE= option. The COLORSTAT= option specifies the statistic to be calculated for the data range of the bar-color gradient. The default is SUM. For existing SAS programs that use the BARCHART statement, if STAT= is used with
COLORRESPONSE= in the BARCHART statement and STAT= specifies a statistic other than SUM, then the bar-chart colors and color statistic might change from those of previous releases. In that case, to restore the original colors and color statistic, set COLORSTAT= in the BARCHART statement to the same statistic that is specified in STAT=.

SAS 9.4M5 changes the behavior of the BARCHART and BARCHARTPARM statement option FILLTYPE=GRADIENT. In earlier releases, GRADIENT fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline. In SAS 9.4M5, GRADIENT is replaced by ALPHAGRADIENT, which has the same behavior as GRADIENT in earlier releases. GRADIENT is changed to an alias of COLORGRADIENT, which fills each bar with an opaque color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in a new option, FILLENDCOLOR=. For existing SAS programs that use the BARCHART or BARCHARTPARM statement FILLTYPE=GRADIENT option, the bar gradients will change. To restore the original bar gradients in that case, specify FILLTYPE=ALPHAGRADIENT.

See BARCHART in “Plot Statement Enhancements” on page xvi.

Programs That Use the DENSITYPLOT Statement

In SAS 9.4, the WEIGHT= option of the KERNEL( ) distribution option in the DENSITYPLOT statement is changed to WEIGHTFUNCTION=. This change enables the addition of the WEIGHT= option in the DENSITYPLOT statement. In SAS 9.4, the WEIGHT= option is not valid in the KERNEL( ) distribution option and results in a syntax error. If your existing SAS programs specify the WEIGHT= option in the KERNEL( ) distribution option, then you must change it to WEIGHTFUNCTION=. See DENSITYPLOT in “Plot Statement Enhancements” on page xvi.

Programs That Use the HEATMAPPARM Statement

SAS 9.4M3 adds options DISCRETEX= and DISCRETEY= to the HEATMAPPARM statement. These options specify whether the X or Y axis is discrete. The default is FALSE. For existing SAS programs that use the HEATMAPPARM statement, if the X axis is discrete, the DISCRETEX=TRUE option must be specified in the HEATMAPPARM statement. Likewise, if the Y axis is discrete, the DISCRETEY=TRUE option must be specified in the HEATMAPPARM statement. Otherwise, the heat map might not be drawn.

Programs That Use the HISTOGRAM or HISTOGRAMPARM Statement

SAS 9.4M5 changes the behavior of the FILLTYPE=GRADIENT option. In earlier releases, GRADIENT fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and
ends with full transparency at the bar baseline. **SAS 9.4M5** replaces option GRADIENT with option ALPHAGRADIENT, which has the same behavior as GRADIENT in earlier releases. GRADIENT is changed to an alias of COLORGRADIENT, which fills each bar with an opaque color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in a new option, FILLENDCOLOR=. For existing SAS programs that use the HISTOGRAM or HISTOGRAMPARM statement FILLTYPE=GRADIENT option, the bar gradients will change. To restore the original bar gradients in that case, specify FILLTYPE=ALPHAGRADIENT.

### Programs That Use the SERIESPLOT Statement

**SAS 9.4M4** changes the behavior of the BREAK=TRUE option to break the series line at missing values of only the Y variable. In earlier SAS releases, the BREAK=TRUE option breaks the series line at missing values of the X or Y variable. If you want your existing SAS programs that specify BREAK=TRUE in a SERIESPLOT statement to break the series line at missing X or Y variable values in **SAS 9.4M4** and later releases, do one of the following:

- In your data, set the Y variable value to missing when the X variable value is missing.
- In the SERIESPLOT statement in your template, use the IFN or IFC function to set the Y variable value to missing when the X variable value is missing, as shown in the following examples.

```sas
seriesplot x=x-var-name y=eval(ifn(x-var-name ne ., y-var-name, .));
seriesplot x=x-var-name y=eval(ifc(x-var-name, y-var-name, ' '));
```

### Programs That Use the WATERFALLCHART Statement

**SAS 9.4M3** adds option COLORSTAT= to the WATERFALLCHART statement. It is enabled by the COLORRESPONSE= option. The COLORSTAT= option specifies the statistic to be calculated for the data range of the color gradient. The default is SUM. For existing SAS programs that use the WATERFALLCHART statement, if STAT= is used with COLORRESPONSE= in the WATERFALLCHART statement and STAT= specifies a statistic other than SUM, then the chart colors and color statistic might change from those of previous releases. In that case, to restore the original colors and color statistic, set COLORSTAT= in the WATERFALLCHART statement to the same statistic that is specified in STAT=. See WATERFALLCHART in “Plot Statement Enhancements” on page xvi.

### Programs That Use the DISPLAY=ALL Option in Plot Statements

**SAS 9.4M5** adds support for fill patterns to band plots, box plots, bubble plots, high-low plots, ellipse plots, and polygon plots. **SAS 9.4M5** also adds fill patterns to the ODS styles. If your SAS program uses the DISPLAY=ALL option in any of these plot
statements, your graph might now display fill patterns in addition to fill color, outline, and so on. If you do not want to use fill patterns in your graph, use the DISPLAY= option in your plot statement to display only the features that you want.

New Plot Statements

The following plot statements are new:

- ANNOTATE draws annotations from annotation instructions that are stored in a SAS data set. It can draw all of the annotations in the data set or a subset only.
- AXISTABLE draws textual values (character or numeric) on the graph at locations that are aligned with the X or Y axis.
- LINECHART creates a summarized line chart that is computed from input data.

SAS 9.4M1 adds the following statement:

- POLYGONPLOT draws one or more polygons from data that is stored in a SAS data set.

SAS 9.4M2 adds the following statement:

- TEXTPLOT displays text values at specific X and Y locations in the graph. For information about subsequent enhancements to this statement, see “Plot Statement Enhancements” on page xvi.

SAS 9.4M3 adds the following statement:

- HEATMAP creates a plot of color-coded rectangles for the response variable of a pair of X and Y variables after it bins the data in two dimensions.

New Legend Statement

SAS 9.4M3 adds the LEGENDTEXTITEMS statement. This statement creates the definition for data-driven text items that can be included in a discrete legend. The items and optional labels are stored in the plot data set.

New Marker Definition Statements

SAS 9.4M1 adds the following statements:

- SYMBOLCHAR defines a marker symbol from a Unicode character value.
- SYMBOLIMAGE defines a marker symbol from a GIF, JPG, or PNG image that is stored in a file.
You can use these statements to define custom marker symbols for your graphs. For information about subsequent enhancements to these statements, see “Marker Definition Statement Enhancements” on page xliii.

New Function

SAS 9.4M1 adds the TYPEOF function. This function returns the type (numeric or character) of a specified column at run time. You can use the TYPEOF function to take specific actions in your template at run time based on the input data type.

New Features for General Use

Color-Priority Graph Data Attribute Rotation

GTL now supports a color-priority rotation pattern for cycling graph data attributes. In the default rotation pattern, graph data attributes rotate as they are defined in the GraphData1–GraphDataN style elements. The new color-priority rotation pattern first cycles through all of the colors while holding the marker symbol, line pattern, or fill pattern constant. When all of the colors have been used, it increments to the next marker symbol, line pattern, or fill pattern, and then repeats the colors.

Color-priority rotation is enabled in one of the following ways:

- the currently active style sets the Graph:attrPriority attribute to COLOR
- the ATTRPRIORITY=COLOR option is specified in the ODS GRAPHICS statement, which overrides the Graph:attrPriority style attribute
- the ATTRPRIORITY=COLOR option is specified in the BEGINGRAPH statement of the template, which overrides the ODS GRAPHICS ATTRPRIORITY= option for all plots in that template

When none of these conditions are true, the default rotation pattern is used. For more information, see “BEGINGRAPH Statement Enhancements” on page xiii.

Template-Level Graph Data Attribute Overrides

GTL provides new options that enable you to override GraphData1–GraphDataN style attributes for all of the plots within a template. They provide an easy way to modify group attributes without having to create a custom ODS style. These options are used in the BEeginGRAPH statement of a template. By using these options, you can override attributes in the GraphData1–GraphDataN style elements by specifying one or more of the following lists for all of the plots within a template:

- a list of colors that replace the graph data colors (fill) in the current style
- a list of contrast colors that replace the graph data contrast colors in the current style
- a list of marker symbols that replace the graph data marker symbols in the current style
- a list of line patterns that replace the graph data line patterns in the current style

In addition, you can specify a priority for the group value attribute cycling. For more information, see “BEGINGRAPH Statement Enhancements” on page xiii.

**Outer Padding**

GTL provides a new OUTERPAD= option that enables you to control the padding around the outside of layouts, legends, titles, footnotes, and text entries. You can use this option to modify the outer padding when the default padding is not sufficient. You can specify a single padding value for all four sides, or you can specify different values for each side.

**Unicode Values in User-Defined Formats**

**SAS 9.4M3** adds support for Unicode values in user-defined formats to ODS Graphics. The Unicode value must be escaped with the (*ESC*) escape sequence, as shown in the following example.

"(*ESC*){unicode beta}"

ODS Graphics does not support the use of a user-defined ODS escape character to escape Unicode values in user-defined formats.

**General Enhancements Supported by Many of the Plots**

The following new features that are supported by many of the plot statements are worth highlighting. The individual plot statements that support these features are identified in “Plot Statement Enhancements” on page xvi.

- The algorithm that is used to place data labels has been improved to more effectively position the data labels in the vicinity of their data markers while avoiding label collisions.
- In certain cases, you can rotate or split data labels, curve labels, and discrete-axis tick values in order to fit them in the available space.
- Subpixel rendering is now supported, which produces smoother curves and more precise bar spacing in plots.
- You can now suppress tips and URLs from individual plots.
BEGINGRAPH Statement Enhancements

- **ATTRPRIORITY=** specifies a priority for cycling of the attributes for group attributes.
- **DATACOLORS=** specifies the list of fill colors that replaces the graph data colors from the GraphData1–GraphDataN style elements.
- **DATACONTRASTCOLORS=** specifies the list of contrast colors that replaces the graph data contrast colors from the GraphData1–GraphDataN style elements.
- **DATALINEPATTERNS=** specifies the list of line patterns that replaces the graph data line patterns from the GraphData1–GraphDataN style elements.
- **DATASKIN=** enhances the visual appearance of all plots in the template that support data skins.
- **DATASYMBOLS=** specifies the list of marker symbols that replaces the graph data marker symbols from the marker symbols that are defined in the GraphData1–GraphDataN style elements.
- **SUBPIXEL=** specifies whether subpixel rendering is used for drawing smooth curved lines or for spacing bars more precisely.

**SAS 9.4M3** has the following enhancements:

- **AXISBREAKSYMBOL=** specifies a symbol to use on the axis lines to indicate a break in the axis.
- **AXISBREAKTYPE=** specifies whether the axis break is indicated in the full display or only on the axis line.
- **AXISLINEEXTENT=** specifies the extent of the axis line for all axes.
- **DISCRETEAXISOFFSETPAD=** specifies whether additional padding is added to the minimum and maximum axis offsets for discrete axes.
- **OPAQUE=** specifies whether the graph background is opaque or transparent.

**SAS 9.4M5** adds option **DATAFILLPATTERNS=**, which specifies a list of fill patterns to replace the graph data fill patterns that are provided by the GraphData1–GraphDataN style elements, if any.

Layout Statement Enhancements

**LAYOUT DATALATTICE** and **LAYOUT DATAPANEL**:

- For cell insets, match-merging is now supported for merging the inset and analysis data. The **DATASCHEMEX** suboption of the **INSETOPTS=** option
specifies whether one-to-one merging or match merging was used to merge the data.

- HEADERLABELDISPLAY= now supports NONE, which suppresses the row and column headings, or the cell headings.
- SORTORDER= specifies the order of the cells along the columns and rows.
- LAYOUT DATALATTICE only:
  - COLUMNDATARANGE=, ROWDATARANGE=, COLUMN2DATARANGE=, and ROW2DATARANGE= now support AUTO, which selects the range automatically, based on the column weight or row weight, and the axis type.
  - COLUMNWEIGHT= specifies how to assign weights to the columns widths.
  - ROWWEIGHT= specifies how to assign weights to the row heights.

SAS 9.4M1 has the following enhancements:

- HEADERPACK= specifies whether the values listed in the cell headers are displayed as a delimited list in a single header cell in order to save space.
- HEADERSEPARATOR= specifies a separator to place between each value in the cell header.
- HEADERSPLITCOUNT= specifies the number of class variables in the cell header after which the cell header wraps to a separate line.

SAS 9.4M2 has the following enhancements:

- For the INSETOPTS= option:
  - CONTENTDISPLAY= specifies whether the variable information displayed in the inset includes the column label and value, or the column value only.
  - SEPARATOR= specifies a new separator for the column label and value.

SAS 9.4M3 adds option HEADERBORDER=, specifies whether a border is drawn around the header cells.

LAYOUT GLOBALLEGEND:

SAS 9.4M1 changes the default padding between the global legend and the plot area (including the axes) to 10 pixels. Use the OUTERPAD= option to adjust the padding, if necessary.

LAYOUT OVERLAY:

INNERMARGIN statement:

- ALIGN= now supports LEFT and RIGHT in addition to TOP and BOTTOM.
- BACKGROUNDCOLOR= specifies the color of the inner margin background.
- OPAQUE= specifies whether the inner margin's background is opaque.

SAS 9.4M1 has the following enhancements:

- SEPARATOR= specifies whether a separating line is drawn between the inner margin and the rest of the layout content.
- SEPARATORATTRS= specifies the attributes of the inner margin separating line.

SAS 9.4M2 has the following enhancements:

- GUTTER= specifies the gap between stacked items in the inner margin.
PAD= specifies the amount of extra space that is added inside the inner margin border.

LAYOUT OVERLAYEQUATED:
- The HEATMAPPARM statement is now supported.
- EQUATETYPE= now supports SQUAREDATA, which specifies that both the X and Y axes have the same range, but can have different tick values.

LAYOUT PROTOTYPE:
- The HEATMAPPARM statement is now supported.
- SAS 9.4M1 enables the INNERMARGIN statement to be used in a LAYOUT PROTOTYPE block.

Legend Statement Enhancements

AXISLEGEND:
- SAS 9.4M2 changes the BORDER= option default to style reference GraphLegendBackground:FrameBorder.

DISCRETELEGEND:
- SAS 9.4M1 has the following enhancements:
  - The default padding between the legend and the plot area (including the axes) is 10 pixels, depending on the context. You can use the OUTERPAD= option to adjust the padding if necessary.
  - ITEMSIZE= specifies the size of specific types of items in a discrete legend.
- SAS 9.4M2 changes the BORDER= option default to style reference GraphLegendBackground:FrameBorder.
- SAS 9.4M3 has the following enhancements:
  - FILLITEMOUTLINE= specifies whether the fill swatches are outlined only when enabled by the contributing statements or are always outlined.
  - The ITEMSIZE= option now supports the following suboptions:
    - FILLASPECTRATIO= specifies the aspect ratio for the fill swatches.
    - FILLHEIGHT= specifies the height of the fill swatches.
    - HEIGHTSCALE= specifies a scaling factor that is to be applied to the fill swatch height.
  - SORTBY= specifies whether text legend items are sorted by label or by text.
  - SAS 9.4M5 adds REVERSEAUTO to option SORTORDER=. REVERSEAUTO presents groups of legend entries in the reverse order in which they are listed in the legend statement.

CONTINUOUSLEGEND:
- EXTRACTSCALE= specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.
- EXTRACTSCALETYPE= specifies whether to extract a named scale or a scientific-notation scale.
- **SAS 9.4M2** changes the `BORDER=` option default to style reference `GraphLegendBackground:FrameBorder`.

- **SAS 9.4M3** adds option `INTEGER=`, which specifies whether only integer tick values are used in the continuous legend.

**LEGENDITEM:**

- `FILLEDOUTLINEDMARKERS=` specifies whether markers are drawn with both fills and outlines.

- **SAS 9.4M3** adds option `FILLDISPLAY=`, which specifies whether the fill swatch for this legend item displays fill only or displays fill and outline.

**MERGEDLEGEND:**

- **SAS 9.4M1** has the following enhancements:
  - `ADDITIONALNAMES=` specifies the name of one or more sources for legend items that are to be added to the merged legend after merging takes place.
  - `ITEMSIZE=` specifies the size of specific types of items in a merged legend.

- **SAS 9.4M2** changes the `BORDER=` option default to style reference `GraphLegendBackground:FrameBorder`.

- **SAS 9.4M5** adds `REVERSEAUTO` to option `SORTORDER=`. `REVERSEAUTO` presents groups of legend entries in the reverse order in which they are listed in the legend statement.

---

**Plot Statement Enhancements**

**AXISTABLE:**

- **SAS 9.4M2** has the following enhancements:
  - `CLASSDISPLAY=` specifies how the class values are displayed.
  - `CLASSORDER=` specifies the order in which the class values are displayed.
  - `CLUSTERWIDTH=` specifies the width of the group clusters as a fraction of the midpoint spacing.
  - `DISPLAY=` now supports `VALUES`, which displays the table values.
  - `DROPONMISSING=` specifies whether the entire axis table is dropped when all of the `VALUE=` column values are missing.
  - `GUTTER=` specifies the gap between rows when a class variable is used.
  - `INCLUDEMISSINGCLASS=` specifies whether missing class values are represented in the table.
  - `LABEL=` specifies the text for the table label.
  - `LABELHALIGN=` specifies the horizontal alignment of the column labels relative to the column width in a Y-axis table.
  - `LABELJUSTIFY=` specifies the justification of the column label, when displayed.
  - `PAD=` specifies the amount of extra space that is added inside the table border.
SHOWMISSING= specifies whether missing values are represented in the table.

TITLE= specifies the text for the table title.

TITLEATTRS= specifies the color and font attributes of the table title.

TITLEHALIGN= specifies the horizontal alignment of the axis table header label relative to the table width for a Y-axis table.

TITLEJUSTIFY= specifies the justification of the title string.

VALUEFORMAT= specifies a SAS format or a user-defined format for the table values.

VALUEHALIGN= specifies the horizontal alignment of the column values relative to the column width in a Y-axis table.

VALUEJUSTIFY= specifies the justification of the values in the axis table.

The following options are replaced and considered deprecated:
- HEADERLABEL= is replaced with TITLE=.
- HEADERLABELATTRS= is replaced with TITLEATTRS=.

The deprecated options are still honored, but the new options are the preferred options.

SAS 9.4M3 adds option TITLEHALIGN=, which specifies the horizontal alignment of the axis table header label relative to the table width for Y-axis tables and X-axis tables.

BANDPLOT:
- TIP= now supports NONE, which suppressed data tips from the plot.
- SAS 9.4M2 adds option ANTIALIAS=, which specifies whether anti-aliasing is turned off for a band plot.
- SAS 9.4M5 has the following enhancements:
  - FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
  - DISPLAY= now supports FILLPATTERN, which displays the band fill pattern.

BARCHART:
- X= is changed to CATEGORY= in order to be consistent with the other GTL plot statements. The X= option is still valid for backward compatibility. However, you should change the X= option to CATEGORY= in your BARCHART statements.

  Note: If you specify X as a data tip role when you change X= to CATEGORY=, then you must also change X to CATEGORY in your data tip options.

- Y= is changed to RESPONSE= in order to be consistent with the other GTL plot statements. The Y= option is still valid for backward compatibility. However, you should change the Y= option to RESPONSE= in your BARCHART statements.

  Note: If you specify Y as a data tip role when you change Y= to RESPONSE=, then you must also change Y to RESPONSE in your data tip options.

- BARLABELFITPOLICY= specifies a policy for avoiding collisions among the bar labels when bar labels are displayed.
BASELINEATTRS= specifies the appearance of the baseline.

The baseline is now drawn by default. To suppress the baseline, use the BASELINEATTRS= option to set the line thickness to 0.

DATALABELFITPOLICY= specifies a policy for avoiding collisions among the bar labels when bar labels are displayed.

STAT= now supports PROPORTION, which displays proportions in the range 0–1.

STAT=PCT now displays percentages in the range 0–100 in order to be consistent with other GTL statements.

Note: In prior SAS releases, the BARCHART statement STAT=PCT option displays proportional values in the range 0–1. To restore the original STAT=PCT results in SAS 9.4 and later releases, specify STAT=PROPORTION in your BARCHART statements instead.

TIP= now supports NONE, which suppressed data tips and URLs from the plot.

SAS 9.4M2 has the following enhancements:

- GROUPORDER= supports REVERSEDATA, which orders the groups within a category in the reverse group-column data order.
- FILLTYPE= specifies whether the fill color is solid or is a gradient that transitions from fully opaque to fully transparent.
- SEGMENTLABEL= specifies whether a label is displayed inside each bar segment.
- SEGMENTLABELATTRS= specifies the text properties of the bar segment label text.
- SEGMENTLABELFITPOLICY= specifies a policy for fitting the bar segment labels within the bar segments.
- SEGMENTLABELFORMAT= specifies the text format used to display the bar segment labels.

SAS 9.4M3 has the following enhancements:

- The BARCHART statement now supports a linear category axis or a time category axis.
- COLORBYFREQ= specifies whether the bar colors are based on the frequency of the category variable.
- COLORMODEL= specifies a color ramp that is to be used with the COLORRESPONSE= option or the COLORBYFREQ= option.
- COLORRESPONSE= specifies the column or range attribute variable that is used to map the bar colors.
- COLORSTAT= specifies the statistic to use for computing the response colors.
- DISPLAYZEROLENGTHBAR= specifies whether zero-length bars are drawn.
- GROUP100= displays the computed response values (FREQ, SUM, or MEAN), normalized to 100%.
- INTERVALBARWIDTH= specifies the width of the bars in an interval bar chart as a ratio of the interval width.
SAS 9.4M5 has the following enhancements:
- **BARLABELFITPOLICY=INSIDEPREFERRED** attempts to place the bar labels in a horizontal bar chart inside the bars.
- **DISPLAYBASELINE=** specifies whether the baseline is displayed.
- **FILLENDCOLOR=color** specifies the end color of the bar fill color gradient.
- **FILLTYPE=ALPHAGRADIENT** replaces **FILLTYPE=GRADIENT** from the previous releases. This option fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.
- **FILLTYPE=COLORGRADIENT** fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the **FILLENDCOLOR=** option.
- **FILLTYPE=GRADIENT** is changed to an alias of **FILLTYPE=COLORGRADIENT**.
- **CONNECTBREAK=** specifies whether the connect line is broken for values that have no observations.

**BARCHARTPARM:**
- **X=** is changed to **CATEGORY=** in order to be consistent with the other GTL plot statements. The **X=** option is still valid for backward compatibility. However, you should change the **X=** option to **CATEGORY=** in your **BARCHARTPARM** statements.
- **Y=** is changed to **RESPONSE=** in order to be consistent with the other GTL plot statements. The **Y=** option is still valid for backward compatibility. However, you should change the **Y=** option to **RESPONSE=** in your **BARCHARTPARM** statements.
- **BASELINEATTRS=** specifies the appearance of the baseline.
  - The baseline is now drawn by default. To suppress the baseline, use the **BASELINEATTRS=** option to set the line thickness to 0.
- **DATALABELFITPOLICY=** specifies a policy for avoiding collisions among the bar labels when bar labels are displayed.
- **DATALABELSPLITCHAR=** specifies one or more characters on which the data labels can be split if needed.
- **DATALABELSPLITCHARDROP=** specifies whether the split characters are included in the data labels.
- **ERRORBARCAPSHAPE=** specifies whether the error bars have a serif cap.
- **TIP=** now supports **NONE**, which suppressed data tips and URLs from the plot.

SAS 9.4M2 has the following enhancements:
- **ERRORBARCAPSHAPE=** defaults to style reference **GraphError:CapStyle**.
- **DATALABELTYPE=** specifies whether the data labels display the **RESPONSE** values or the values of the column specified by the **DATALABEL=** option.
- **FILLTYPE=** specifies whether the fill color is solid or is a gradient that transitions from fully opaque to fully transparent.
- **GROUPORDER=** supports **REVERSEDATA**, which orders the groups within a category in the reverse group-column data order.
- SEGMENTLABELATTRS= specifies the text properties of the bar segment label text.
- SEGMENTLABELFITPOLICY= specifies a policy for fitting the bar segment labels within the bar segments.
- SEGMENTLABELFORMAT= specifies the text format used to display the bar segment labels.
- SEGMENTLABELTYPE= specifies whether a label is displayed inside each bar segment.

**SAS 9.4M3** has the following enhancements:
- The BARCHARTPARM statement now supports a linear category axis or a time category axis.
- COLORMODEL= specifies a color ramp that is to be used with the COLORRESPONSE= option.
- COLORRESPONSE= specifies the column or range attribute variable that is used to map the bar colors.
- DISPLAYZEROLENGTHBAR= specifies whether zero-length bars are drawn.
- GROUP100= displays the computed values normalized to 100%.
- INTERVALBARWIDTH= specifies the width of the bars as a ratio of the interval width.

**SAS 9.4M5** has the following enhancements:
- DATALABELFITPOLICY=INSIDEPREFERRED attempts to place the data labels in a horizontal bar chart inside the bars.
- DISPLAYBASELINE= specifies whether the baseline is displayed.
- FILLENDCOLOR=color specifies the end color of the bar fill color gradient.
- FILLTYPE=ALPHAGRADIENT replaces FILLTYPE=GRADIENT from the previous releases. This option fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.
- FILLTYPE=COLORGRADIENT fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the FILLENDCOLOR= option.
- FILLTYPE=GRADIENT is changed to an alias of FILLTYPE=COLORGRADIENT.
- CONNECTBREAK= specifies whether the connect line is broken for values that have no observations.
- ERRORBARCAPSCALE= specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

**BLOCKPLOT:**
- BLOCKLABEL= specifies alternative text to display for the internal block text values.
- VALUEFITPOLICY= now supports the SPLIT, SPLITALWAYS, and NONE policies.
- VALUESPLITCHAR= specifies one or more characters on which the values can be split if needed.
VALUESPLITCHARDROP= specifies whether the split characters are included in the displayed values.

BOXPLOT:

- CAPSHAPE= now supports NONE, which specifies that no shape is displayed at the ends of the box whiskers.
- DATALABELSPLIT= specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.
- DATASKIN= enhances the visual appearance of the filled boxes.
- DISPLAYSTATS= specifies the statistics that are displayed for each box.
- TIP= now supports NONE, which suppressed data tips from the plot.
- WEIGHT= specifies a column that contains a statistics calculation for each observation of the input data object.

SAS 9.4M1 has the following enhancements:

- WHISKERPERCENTILE= specifies the whisker length in percentile units.
- DISPLAYSTATS= now displays the DATAMAX, DATAMIN, and SUMWGT statistics.
- LEGENDLABEL= now defaults to the Y= column label or name.

SAS 9.4M2 adds option GROUPORDER=REVERSEDATA, which orders the groups within a category in the reverse group-column data order.

SAS 9.4M3 modifies the BOXPLOT statement to contribute its mean markers to a discrete legend when the BOXPLOT DISPLAY= option includes MEAN, and TYPE=MARKER is specified in the DISCRETELEGEND statement.

SAS 9.4M5 has the following enhancements:

- DISPLAYSTATS= can be used with both vertical and horizontal box plots.
- CONNECTBREAK= specifies whether the connect line is broken for values that have no observations.
- CAPSCALE= specifies a positive number to use as a multiplier for determining the default width of the whisker caps.
- FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
- DISPLAY= now supports FILLPATTERN, which displays the box fill pattern.

BOXPLOTPARM:

- CAPSHAPE= now supports NONE, which specifies that no shape is displayed at the ends of the box whiskers.
- DATALABELSPLIT= specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.

DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.

DATASKIN= enhances the visual appearance of the filled boxes.

TIP= now supports NONE, which suppresses data tips from the plot.

SAS 9.4M1 has the following enhancements:

- DISPLAYSTATS= now displays the DATAMAX, DATAMIN, and SUMWGT statistics.
- LEGENDLABEL= now defaults to the Y= column label or name.
- URL= specifies an HTML page that is displayed when a box or an outlier marker is selected.

SAS 9.4M2 adds option GROUPORDER=REVERSEDATA, which orders the groups within a category in the reverse group-column data order.

SAS 9.4M3 modifies the BOXPLOTPARM statement to contribute its mean markers to a discrete legend when the BOXPLOTPARM DISPLAY= option includes MEAN, and TYPE=MARKER is specified in the DISCRETELEGEND statement.

SAS 9.4M5 has the following enhancements:

- DISPLAYSTATS= can be used with both vertical and horizontal box plots.
- CONNECTBREAK= specifies whether the connect line is broken for values that have no observations.
- CAPSCALE= specifies a positive number to use as a multiplier for determining the default width of the whisker caps.
- FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
- DISPLAY= now supports FILLPATTERN, which displays the box fill pattern.

BUBBLEPLOT:

DATALABELSSPLIT= specifies whether to split the data labels at specified split characters.

DATALABELSSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.

DATALABELSSPLITCHARDROP= specifies whether the split characters are included in the data labels.

DATALABELSSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.

TIP= now supports NONE, which suppresses data tips and URLs from the plot.

SAS 9.4M3 adds option DRAWORDER=, which specifies whether the bubbles are drawn according to bubble size or according to data order.

SAS 9.4M5 has the following enhancements:

- FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
- DISPLAY= now supports FILLPATTERN, which displays the bubble fill pattern.

CONTOURPLOTPARM:
LEVELS= specifies a list of contour level values.

DENDROGRAM:
TIP= now supports NONE, which suppresses data tips and URLs from the plot.

DENSITYPLOT:
CURVELABELSPLIT= specifies whether to split the curve label at the specified split characters.
CURVELABELSPLITCHAR= specifies one or more characters on which the curve label can be split if needed.
CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the curve label text.
CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the curve label block.
WEIGHT= specifies a column that contains a density-curve calculation a priori weight for each observation of the input data object.

The KERNEL() distribution option WEIGHT= is changed to WEIGHTFUNCTION=. This change enables the addition of the WEIGHT= option in the DENSITYPLOT statement.

Note: The WEIGHT= option is not valid in the KERNEL() distribution option in SAS 9.4. If you used the WEIGHT= option in the KERNEL() distribution option in prior SAS releases, then you must change it to WEIGHTFUNCTION= in SAS 9.4 and later releases.

SAS 9.4M2 has the following enhancements:
GROUP= creates a separate density curve for each unique group value of the specified column.
INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot.

DROPLINE:
DATASKIN= enhances the appearance of the drop line.
SAS 9.4M1 adds option DROPTO=BOTH, which draws one or more drop lines to both the X and Y axes.

ELLIPSE:
SAS 9.4M2 has the following enhancements:
GROUP= creates a separate ellipse for each unique group value of the specified column.
INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot.
SAS 9.4M5 has the following enhancements:
FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
DISPLAY= now supports FILLPATTERN, which displays the band fill pattern.

ELLIPSEPARM:
SAS 9.4M5 has the following enhancements:
FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
DISPLAY= now supports FILLPATTERN, which displays the ellipse fill pattern.

FRINGE PLOT:
GROUP= creates a distinct set of lines for each unique group value in the specified column.
INCLUDEMISSINGGROUP= specifies whether missing values in the group column are included in the plot.
INDEX= specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.
TIP= now supports NONE, which suppresses data tips from the plot.

HEATMAP PARM:
URL= specifies an HTML page to display when a rectangle is selected.
TIP= now supports NONE, which suppresses data tips and URLs from the plot.

SAS 9.4M2 adds option INCLUDEMISSINGCOLOR=, which specifies whether missing values of the color-group variable or the color-response variable are included in the plot.

SAS 9.4M3 has the following enhancements:
- DISCRETEX= specifies whether the X axis is discrete when X= specifies a numeric column.
- DISCRETEY= specifies whether the Y axis is discrete when Y= specifies a numeric column.

HIGHLOW PLOT:
CLIPCAP= specifies whether a special clip cap is displayed to indicate where clipping occurred.
CLIPCAPSHAPE= specifies the shape of the arrowhead on the clipped end of a line.
DATASKIN= enhances the visual appearance of the high-low chart filled bars or lines.
ENDCAPDISPLAYPOLICY= specifies the policy for displaying end caps when end caps are present.
TIP= now supports NONE, which suppresses data tips from the plot.

SAS 9.4M2 adds option GROUPORDER=REVERSEDATA, which orders the groups within a category in the reverse group-column data order.

SAS 9.4M3 has the following enhancements:
- COLORMODEL= specifies a color ramp that is used with the COLORRESPONSE= option.
- COLORRESPONSE= specifies the column or range attribute variable that is used to map the bar or line colors.

HISTOGRAM:
DATALABELTYPE= specifies the statistic to display at the end of each bar.
DATASKIN= enhances the visual appearance of the filled bars.
WEIGHT= specifies a column that contains a bin-width calculation \textit{a priori} weight for each observation of the input data object.

SAS 9.4M1 has the following enhancements:
- The number of bins is limited to approximately 10,000. When the limit is exceeded, SAS automatically adjusts the NBINS= or BINWIDTH= value to set the number of bins to about 10,000.
- DISPLAY= now supports FILLPATTERN.
- FILLPATTERNATTRS= specifies the appearance of the pattern-filled bar area.

SAS 9.4M2 has the following enhancements:
- FILLTYPE= specifies whether the fill color is solid or is a gradient that transitions from fully opaque to fully transparent.
- GROUP= creates a separate bar segment or bar for each unique group value of the specified column.
- INCLUDEMISSINGGROUP= specifies whether missing values of the group variable are included in the plot.
- The OUTLINEATTRS= option defaults are now consistent with that of BARCHART.

SAS 9.4M5 has the following enhancements:
- DATALABELATTRS= specifies the color and font attributes of the bar labels.
- FILLENDCOLOR=\emph{color} specifies the end color of the bar fill color gradient.
- FILLTYPE=ALPHAGRADIENT replaces FILLTYPE=GRADIENT from the previous releases. This option fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.
- FILLTYPE=COLORGRADIENT fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the FILLENDCOLOR= option.
- FILLTYPE=GRADIENT is changed to an alias of FILLTYPE=COLORGRADIENT.

HISTOGRAMPARM:
- DATALABELFITPOLICY= specifies a policy for avoiding collisions among the bin labels when bin labels are displayed.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATASKIN= enhances the visual appearance of the filled bars.
- TIP= now supports NONE, which suppresses data tips from the plot.

SAS 9.4M1 has the following enhancements:
- DISPLAY= now supports FILLPATTERN.
- FILLPATTERNATTRS= specifies the appearance of the pattern-filled bar area.
SAS 9.4M2 adds option FILLTYPE=, which specifies whether the fill color is solid or is a gradient that transitions from fully opaque to fully transparent.

SAS 9.4M5 has the following enhancements:
- FILLENDCOLOR=color specifies the end color of the bar fill color gradient.
- FILLTYPE=ALPHAGRADIENT replaces FILLTYPE=GRADIENT from the previous releases. This option fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.
- FILLTYPE=COLORGRADIENT fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the FILLENDCOLOR= option.
- FILLTYPE=GRADIENT is changed to an alias of FILLTYPE=COLORGRADIENT.

LINECHART:
SAS 9.4M2 adds option GROUPORDER=REVERSEDATA, which orders the groups within a category in the reverse group-column data order.

SAS 9.4M3 has the following enhancements:
- COLORMODEL= specifies a color ramp that is used with the COLORRESPONSE= option.
- COLORRESPONSE= specifies the column or range attribute variable that is used to map the line, marker, and fill colors.

LINEPARM:
- CURVELABELSPLIT= specifies whether to split the line label at the specified split characters.
- CURVELABELSPLITCHAR= specifies one or more characters on which the line label can be split if needed.
- CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the line label text.
- CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the line label block.

LOESSPLOT:
- CURVELABELSPLIT= specifies whether to split the curve label at the specified split characters.
- CURVELABELSPLITCHAR= specifies one or more characters on which the curve label can be split if needed.
- CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the curve label text.
- CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the curve label block.

MODELBAND:
SAS 9.4M2 has the following enhancements:
- ANTIALIAS= specifies whether anti-aliasing is turned off for a model band plot.
A confidence band that depicts confidence limits for individual predicted values (CLI) for a weighted spline plot or regression plot is now displayed as a high-low chart instead of a band.

**NEEDLEPLOT:**
- `BASELINEATTRS=` specifies the appearance of the baseline. This option enables you to suppress the baseline by setting the line thickness to 0.
- `DATALABELSPLIT=` specifies whether to split the data labels at specified split characters.
- `DATALABELSPLITCHAR=` specifies one or more characters on which the data labels can be split if needed.
- `DATALABELSPLITCHARDROP=` specifies whether the split characters are included in the data labels.
- `DATALABELSPLITJUSTIFY=` specifies the justification of the strings that are inside the data label blocks.
- `DATASKIN=` enhances the appearance of the needle plot lines.
- `MARKERATTRS=` now supports transparency.
- `TIP=` now supports `NONE`, which suppresses data tips and URLs from the plot.
- **SAS 9.4M2** adds option `GROUPORDER=REVERSEDATA`, which orders the groups within a category in the reverse group-column data order.
- **SAS 9.4M5** adds option `DISPLAYBASELINE=`, which specifies whether the baseline is displayed.

**PBSPLINEPLOT:**
- `CURVELABELSPLIT=` specifies whether to split the curve label at the specified split characters.
- `CURVELABELSPLITCHAR=` specifies one or more characters on which the curve label can be split if needed.
- `CURVELABELSPLITCHARDROP=` specifies whether the split characters are included in the curve label text.
- `CURVELABELSPLITJUSTIFY=` specifies the justification of the strings that are inside the curve label block.
- **SAS 9.4M2** changes the range of the `PBSPLINEPLOT DEGREE=` option from 0–174 to 0–10.

**PIECHART:**
- `TIP=` now supports `NONE`, which suppresses data tips and URLs from the plot.
- **SAS 9.4M1** adds option `CENTERFIRSTSLICE=`, which specifies whether the first pie slice is centered on the starting angle or starts on it.
- **SAS 9.4M3** adds sub option `TRANSPARENCY=` to the `PIECHART` statement `FILLATTRS=` option. The new `TRANSPARENCY=` sub option sets the transparency of the other slice unless transparency is specified in the `OTHERSLICEOPTS=` option.

**POLYGONPLOT:**
- **SAS 9.4M2** has the following enhancements:
  - `ANTIALIAS=` specifies whether anti-aliasing is turned off for a polygon plot.
- BACKLIGHT= specifies a back-light effect for the polygon label text.

SAS 9.4M5 has the following enhancements:
- FILLPATTERNATTRS= specifies the appearance of the pattern-filled areas.
- DISPLAY= now supports FILLPATTERN, which displays the band fill pattern.

REFERENCELINE:
- CURVELABELSPLIT= specifies whether to split the reference line label at the specified split characters.
- CURVELABELSPLITCHAR= specifies one or more characters on which the reference line label can be split if needed.
- CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the reference line label text.
- CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the reference line label block.
- DATASKIN= enhances the visual appearance of the reference line.

SAS 9.4M5 has the following enhancements:
- DISCRETELINETHICKNESS= specifies the line thickness as a fraction of the midpoint spacing.
- USEDISCRETELINETHICKNESS= specifies that the line thickness is based on a fraction of the midpoint spacing that is set by the DISCRETELINETHICKNESS= option.

REGRESSIONPLOT:
- CURVELABELSPLIT= specifies whether to split the regression line label at the specified split characters.
- CURVELABELSPLITCHAR= specifies one or more characters on which the regression line label can be split if needed.
- CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the regression line label text.
- CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the regression line label block.

SAS 9.4M2 changes the range of the DEGREE= regression option from 0–174 to 1–10.

SCATTERPLOT:
- CLUSTERAXIS= specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.
- DATALABELSPLIT= specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.
Enhancements to SAS 9.4 Statements

- **ERRORBARCAPSHAPE=** specifies whether the error bars have a serif cap. **SAS 9.4M2** changes the default of this option to style reference GraphError:CapStyle.

- **FILLEDOUTLINEDMARKERS=** specifies whether markers are drawn with both fills and outlines.

- **JITTER=** specifies whether to jitter data markers.

- **JITTEROPTS=** specifies options for managing jittering.

- **LABELSTRIP=** specifies whether leading and trailing blanks are stripped from marker characters or fixed-position data labels before they are displayed in the plot.

- **MARKERATTRS=** now supports transparency.

- **MARKERCHARACTERPOSITION=** specifies the justification of the marker characters.

- **MARKERFILLATTRS=** specifies the appearance of the filled markers.

- **MARKEROUTLINEATTRS=** specifies the appearance of the marker outlines.

- **TIP=** now supports NONE, which suppresses data tips and URLs from the plot.

- **SAS 9.4M1** adds option OUTLINEDMARKERCHARACTERS=, which specifies whether backlighting or a drop shadow is applied to the characters that are used as marker symbols in order to enhance their appearance in the graph.

- **SAS 9.4M2** has the following enhancements:
  - **CONTRIBUTEOFFSETS=** specifies whether the plot's space requirements contribute to the calculation of the axis offsets.
  - The **GROUPORDER=** option supports REVERSEDATA, which orders the groups within a category in the reverse group-column data order.
  - The following options are replaced and considered deprecated:
    - **MARKERCOLORGRADIENT=** is replaced with **COLORRESPONSE=**.
    - **MARKERSIZERESPONSE=** is replaced with **SIZERESPONSE=**.
    - **MARKERSIZEMAX=** is replaced with **SIZEMAX=**.
    - **MARKERSIZEMIN=** is replaced with **SIZEMIN=**.
      The new options are functionally the same as the deprecated options and are more consistent with the other plot statements. The deprecated options are still honored, but the new options are the preferred options.
  - The **OUTLINEDMARKERCHARACTERS=** option is deprecated. It is still honored, but the **TEXTPLOT** statement is now the preferred method for creating scatter plots using text markers.

- **SAS 9.4M3** adds option **SUBPIXEL=**, which specifies whether subpixel rendering is used for image output when the scatter plot is rendered.

- **SAS 9.4M5** adds option **ERRORBARCAPSCALE=**, which specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

### SCATTERPLOTMATRIX:

- **DATALABELSPLIT=** specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.
- DATASKIN= enhances the visual appearance of the plot markers.
- ELLIPSE= now supports the CLIP= option, which specifies whether clipped confidence ellipses are included in the plot.
- LABELSTRIP= specifies whether leading and trailing blanks are stripped from marker characters or fixed-position data labels before they are displayed in the plot.
- MARKERATTRS= now supports transparency.
- MARKERCHARACTERPOSITION= specifies the justification of the marker characters.
- MATRIXTYPE= specifies whether to display the full matrix, or just the upper or lower triangle of the matrix.
- TIP= now supports NONE, which suppresses data tips from the plot.

**SAS 9.4M2** replaces option MARKERCOLORGRADIENT= with the COLORRESPONSE= option. Option MARKERCOLORGRADIENT= is considered deprecated. The COLORRESPONSE= option is more consistent with the other plot statements. The MARKERCOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is the preferred option.

**SAS 9.4M3** adds option SUBPIXEL=, which specifies whether subpixel rendering is used for image output when the scatter plots are rendered.

### SERIESPLOT:
- CLUSTERAXIS= specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.
- DATALABELSPLIT= specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.
- CURVELABELSPLIT= specifies whether to split the series line label at the specified split characters.
- CURVELABELSPLITCHAR= specifies one or more characters on which the series line label can be split if needed.
- CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the series line label text.
- CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the series line label block.
- DATASKIN= enhances the visual appearance of the series plot lines.
FILLEDOUTLINEDMARKERS= specifies whether markers are drawn with both fills and outlines.

MARKERATTRS= now supports transparency.

MARKERFILLATTRS= specifies the appearance of the filled markers.

MARKEROUTLINEATTRS= specifies the appearance of the marker outlines.

TIP= now supports NONE, which suppresses data tips and URLs from the plot.

SAS 9.4M2 has the following enhancements:

- The GROUPORDER= option supports REVERSEDATA, which orders the groups within a category in the reverse group-column data order.
- LINECOLORGROUP= specifies a column that determines the line colors for a grouped plot independently of the GROUP= column.
- LINEPATTERNGROUP= specifies a column that determines the line patterns for a grouped plot independently of the GROUP= column.
- MARKERCOLORGROUP= specifies a column that determines the marker colors for a grouped plot independently of the GROUP= column.
- MARKERSYMBOLGROUP= specifies a column that determines the marker symbols for a grouped plot independently of the GROUP= column.

SAS 9.4M3 has the following enhancements:

- ARROWHEADPOSITION= specifies a position for arrowheads.
- ARROWHEADSCALE= specifies an arrowhead scale factor based on the thickness of the arrow line.
- ARROWHEADSHAPE= specifies a shape for arrowheads.
- COLORMODEL= specifies a color ramp that is to be used with the COLORRESPONSE= option.
- COLORRESPONSE= specifies the column or range attribute variable that is used to map the line and marker colors.
- LINETHICKNESSMAX= specifies the maximum line thickness when a response variable is used to determine the line thickness.
- LINETHICKNESSMAXRESPONSE= specifies the response value that corresponds to the maximum line thickness.
- LINETHICKNESSMIN= specifies the minimum line thickness when a response variable is used to determine the line thickness.
- LINETHICKNESSRESPONSE= specifies a response column or range attribute variable that is used to map a line thickness to each group value.
- SPLINEPOINTS= specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.
- SPLINETYPE= specifies the type of spline interpolation that is used to draw the series line.
- The LINECOLORGROUP=, LINEPATTERNGROUP=, MARKERCOLORGROUP=, and MARKERSYMBOLGROUP= options now support a discrete attribute map variable.

SAS 9.4M4 has the following enhancement and change:

- LINEJOIN= specifies how to treat the vertices of the series line.
BREAK= breaks the series line at missing values of the Y variable only. In earlier SAS releases, the BREAK= option breaks the series line at missing values of the X or Y variable.

Note: If you want to break the series line at missing X or Y variable values in SAS 9.4M4 and later releases, set the Y variable value to missing when the X variable value is missing. See “Programs That Use the SERIESPLOT Statement” on page ix.

SAS 9.4M5 has the following enhancements:

- ERRORBARATTRS= specifies the attributes of the error bars that are associated with the data points.
- ERRORBARCAPSHAPE= specifies whether the error bars have a serif cap.
- XERRORLOWER= specifies values for the lower endpoints on the X error bars.
- XERRORUPPER= specifies values for the upper endpoints on the X error bars.
- YERRORLOWER= specifies values for the lower endpoints on the Y error bars.
- YERRORUPPER= specifies values for the upper endpoints on the Y error bars.
- ERRORBARCAPSCALE= specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

STEPPLOT:

- CLUSTERAXIS= specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.
- DATALABELSPLIT= specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.
- DATASKIN= enhances the appearance of the step plot lines.
- ERRORBARCAPSHAPE= specifies whether the error bars have a serif cap.
- CURVELABELSPLIT= specifies whether to split the step line label at the specified split characters.
- CURVELABELSPLITCHAR= specifies one or more characters on which the step line label can be split if needed.
- CURVELABELSPLITCHARDROP= specifies whether the split characters are included in the step line label text.
- CURVELABELSPLITJUSTIFY= specifies the justification of the strings that are inside the step line label block.
- FILLEDOUTLINEDMARKERS= specifies whether markers are drawn with both fills and outlines.
- MARKERATTRS= now supports transparency.
- MARKERFILLATTRS= specifies the appearance of the filled markers.
- MARKEROUTLINEATTRS= specifies the appearance of the marker outlines.
- TIP= now supports NONE, which suppresses data tips and URLs from the plot.

SAS 9.4M2 has the following enhancements:
- ERRORBARCAPSHAPE= defaults to style reference GraphError:CapStyle.
- The GROUPORDER= option supports REVERSEDATA, which orders the groups within a category in the reverse group-column data order.

SAS 9.4M3 has the following enhancements:
- ARROWHEADPOSITION= specifies a position for arrowheads.
- ARROWHEADSCALE= specifies an arrowhead scale factor based on the thickness of the arrow line.
- ARROWHEADSHAPE= specifies a shape for arrowheads.
- LINETHICKNESSMAX= specifies the maximum line thickness when a response variable is used to determine the line thickness.
- LINETHICKNESSMAXRESPONSE= specifies the response value that corresponds to the maximum line thickness.
- LINETHICKNESSMIN= specifies the minimum line thickness when a response variable is used to determine the line thickness.
- LINETHICKNESSRESPONSE= specifies a response column or range attribute variable that is used to map a line thickness to each group value.

SAS 9.4M5 adds option ERRORBARCAPSCALE=, which specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

SURFACEPLOTPARM:
- SAS 9.4M2 replaces option SURFACECOLORGRADIENT= with the COLORRESPONSE= option. The SURFACECOLORGRADIENT= option is considered deprecated. The COLORRESPONSE= option is more consistent with the other plot statements. The SURFACECOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is the preferred option.

TEXTPLOT:
- SAS 9.4M3 has the following experimental options:
  - OUTFILE= specifies a file for storing information about the text bounding-box for each text value in the column specified in the OUTID= option.
  - OUTID= specifies a column that contains text values to write to the file specified in the OUTFILE= option.

SAS 9.4M5 has the following enhancements:
- POSITION= now accepts the name of a keyword column.
- POSITIONOFFSETX= specifies the positive or negative X offset for an individual text item.
POSITIONOFFSETY= specifies the positive or negative Y offset for an individual text item.

VECTORPLOT:
- DATALABELSPLIT= specifies whether to split the data labels at specified split characters.
- DATALABELSPLITCHAR= specifies one or more characters on which the data labels can be split if needed.
- DATALABELSPLITCHARDROP= specifies whether the split characters are included in the data labels.
- DATALABELSPLITJUSTIFY= specifies the justification of the strings that are inside the data label blocks.
- DATASKIN= enhances the appearance of the vector plot lines.
- TIP= now supports NONE, which suppresses data tips from the plot.

SAS 9.4M3 has the following enhancements:
- COLORMODEL= specifies a color ramp that is used with the COLORRESPONSE= option.
- COLORRESPONSE= specifies the column or range attribute variable that is used to map the line colors.
- LINETHICKNESSMAX= specifies the maximum line thickness when a response variable is used to determine the line thickness.
- LINETHICKNESSMAXRESPONSE= specifies the response value that corresponds to the maximum line thickness.
- LINETHICKNESSMIN= specifies the minimum line thickness when a response variable is used to determine the line thickness.
- LINETHICKNESSRESPONSE= specifies a response column or range attribute variable that is used to map a line thickness to each group value.

WATERFALLCHART:
- BARLABELFITPOLICY= specifies a policy for avoiding collisions among the bar labels when bar labels are displayed.
- BASELINEATTRS= specifies the appearance of the baseline. This option enables you to suppress the baseline by setting the line thickness to 0.
- TIP= now supports NONE, which suppresses data tips and URLs from the plot.

SAS 9.4M3 adds option COLORSTAT=, which specifies the statistic to use for computing the response colors.

SAS 9.4M5 adds option CONNECTBREAK=, which specifies whether the connect line is broken for values that have no observations.

Text Statement Enhancements

ENTRY:
- SAS 9.4M1 removes the 512-character limit on the length of the entry text.

ENTRYFOOTNOTE:
HALIGNCENTER= specifies whether the footnote is centered automatically by the system or is always centered in the graph area.

SAS 9.4M1 removes the 512-character limit on the length of the footnote text.

ENTRYTITLE:

HALIGNCENTER= specifies whether the title is centered automatically by the system or is always centered in the graph area.

SAS 9.4M1 removes the 512-character limit on the length of the title text.

Draw Statement Enhancements

SAS 9.4M1 adds option URL= to the BEGINPOLYGON, BEGINPOLYLINE, DRAWTEXT, DRAWLNE, DRAWARROW, DRAWRECTANGLE, DRAWIMAGE, and DRAWOVAL statements. The new URL= option specifies an HTML page that is displayed when the output of the draw statement is selected.

Axis Statement Enhancements

LAYOUT OVERLAY:

LABELFITPOLICY= specifies a policy for fitting axis labels in the available space.

LABELPOSITION= specifies the position of the axis label.

LABELSPLITCHAR= specifies one or more characters on which the axis labels can be split if needed.

LABELSPLITCHARDROP= specifies whether the split characters are included in the axis labels.

LABELSPLITJUSTIFY= specifies the justification of the strings that are inside the axis label blocks.

TICKVALUEHALIGN= specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.

TICKVALUEVALIGN= specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.

SAS 9.4M3 adds option LINEEXTENT=, which specifies the extent of the axis line.

SAS 9.4M5 adds option LABELFITPOLICY=STACKED, which displays the Y and Y2 axis label vertically as stacked letters.

DISCRETEOPTS= supports the following new features for discrete axes:

- TICKDISPLAYLIST= specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.
- TICKVALUEFITPOLICY= supports new fit policies: ROTATE, ROTATEALWAYS, ROTATEALWAYSDROP, SPLIT, SPLITALWAYS, SPLITALWAYSTHIN, SPLITTHIN, and SPLITROTATE.
- **TICKVALUELIST**= specifies the list of tick values that are displayed on the axis.
- **TICKVALUEROTATION**= specifies how the tick values are rotated on the X and X2 axes.
- **TICKVALUESPLITCHAR**= specifies a list of characters on which the tick values can be split if needed.
- **TICKVALUESPLITJUSTIFY**= specifies justification of the strings that are inside the tick value block.
- **TICKVALUESPLITCHARDROP**= specifies whether the split characters are included in the displayed tick values.

**SAS 9.4M3** adds option **TICKVALUEFORMAT**=, which specifies how to format the values for major tick marks.

**SAS 9.4M5** has the following enhancements:

- **TICKVALUEROTATION=DIAGONAL2** rotates the tick values to a –45-degree diagonal position.
- **TICKVALUEFITPOLICY=STACKEDALWAYS** always displays the tick values vertically as stacked letters.
- **TICKVALUEFITPOLICY=STACKEDALWAYSTHIN** is the same as STACKEDALWAYS, except that thinning is performed when the tick values do not fit the available space.

**LINEAROPTS**= supports the following new features for linear axes:

- **MINORGRID**= specifies whether grid lines are displayed at the minor tick values.
- **MINORGRIDATTRS**= specifies the attributes of the minor grid lines.
- **MINORTICKCOUNT**= specifies the number of minor ticks that are displayed on the axis.
- **MINORTICKS**= specifies whether the minor tick marks are displayed on the axis.
- **TICKDISPLAYLIST**= specifies the text that is displayed for the tick values that are defined in the **TICKVALUELIST**= option.
- **TICKVALUEFITPOLICY**= now supports policies **NONE** and **ROTATEALWAYS**.
- **TICKVALUEFORMAT**= now supports **EXTRACTSCALETYPE**=, which enables you to specify the type of scale that you want to extract.
- **TICKVALUEROTATION**= specifies how the tick values are rotated on the X and X2 axes.

**SAS 9.4M1** adds option **INCLUDERANGES**=, which specifies the ranges for a broken axis.

**SAS 9.4M2** has the following enhancements:

- The **MINORGRID**= option defaults to style reference **GraphMinorGridLines:DisplayOpts**.
- The **MINORGRIDATTRS**= option defaults to style element **GraphMinorGridLines** in order to visually contrast the major and minor grid lines.
The MINORTICKCOUNT= option for linear axes defaults to one minor tick and two intervals.

- **SAS 9.4M5** adds option TICKVALUEROTATION=DIAGONAL2, which rotates the tick values to a –45-degree diagonal position.

- **LOGOPTS=** supports the following new features for log axes:
  - MINORGRID= specifies whether grid lines are displayed at the minor tick values.
  - MINORGRIDATTRS= specifies the attributes of the minor grid lines.
  - MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
  - TICKVALUELIST= specifies the tick values for a log axis as a space-separated list.
  - TICKVALUEPRIORITY= specifies whether the TICKVALUELIST= specification can extend the axis data range.
  - VALUETYPES= specifies how the VIEWMIN=, VIEWMAX=, and TICKVALUELIST= option values are interpreted.

- **SAS 9.4M2** has the following enhancements:
  - The MINORGRID= option defaults to style reference GraphMinorGridLines:DisplayOpts.
  - The MINORGRIDATTRS= option defaults to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.

- **SAS 9.4M3** adds option TICKVALUEFORMAT=, which specifies how to format the values for major tick marks.

- **TIMEOPTS=** supports the following new features for time axes:
  - MINORGRID= specifies whether grid lines are displayed at the minor tick values.
  - MINORGRIDATTRS= specifies the attributes of the minor grid lines.
  - MINORTICKINTERVAL= specifies the time interval between minor ticks.
  - TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
  - TICKVALUEROTATION= specifies how the tick values are rotated on the X and X2 axes.

- **SAS 9.4M1** adds option INCLUDERANGES=, which specifies the ranges for a broken axis.

- **SAS 9.4M2** has the following enhancements:
  - The MINORGRID= option defaults to style reference GraphMinorGridLines:DisplayOpts.
  - The MINORGRIDATTRS= option defaults to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.

- **SAS 9.4M3** adds option INTERVALMULTIPLIER=, which specifies a multiplier to apply to the time interval that is in effect for the axis.
SAS 9.4M5 adds option TICKVALUEROTATION=DIAGONAL2, which rotates the tick values to a –45-degree diagonal position.

LAYOUT OVERLAY3D:
- LINEAROPTS= supports the following new features for linear axes:
  - MINORGRID= specifies whether grid lines are displayed at the minor tick values.
  - MINORGRIDATTRS= specifies the attributes of the minor grid lines.
  - MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
  - MINORTICKS= specifies whether the minor tick marks are displayed on the axis.
  - TICKDISPLAYLIST= specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.
  - TICKVALUEFORMAT= now supports EXTRACTSCALETYPE=, which enables you to specify the type of scale that you want to extract.
  - TICKVALUEPRIORITY= specifies whether the TICKVALUELIST= specification can extend the axis data range.
- SAS 9.4M2 changes the MINORGRIDATTRS= option default to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.

TIMEOPTS= supports the following new features for time axes:
- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKINTERVAL= specifies the time interval between minor ticks.
- SAS 9.4M2 changes the MINORGRIDATTRS= option default to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.
- SAS 9.4M3 adds option INTERVALMULTIPLIER=, which specifies a multiplier to apply to the time interval that is in effect for the axis.

LAYOUT OVERLAYEREQUATED:
- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
- MINORTICKS= specifies whether the minor tick marks are displayed on the axis.
- TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
- TICKVALUEFORMAT= now supports EXTRACTSCALETYPE=, which enables you to specify the type of scale that you want to extract.
- VIEWMAX= specifies the maximum data value to include in the display (the value might be adjusted by the threshold calculation).
- VIEWMIN= specifies the minimum data value to include in the display (the value might be adjusted by the threshold calculation).
- **SAS 9.4M3** adds option **LINEEXTENT=**, which specifies the extent of the axis lines.

**LAYOUT LATTICE:**
- **LABELFITPOLICY=** specifies a policy for fitting axis labels in the available space.
- **LABELPOSITION=** specifies the position of the axis label.
- **LABELSPLITCHAR=** specifies one or more characters on which the axis labels can be split if needed.
- **LABELSPLITCHARDROP=** specifies whether the split characters are included in the axis labels.
- **LABELSPLITJUSTIFY=** specifies the justification of the strings that are inside the axis label blocks.
- **TICKVALUEHALIGN=** specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.
- **TICKVALUEVALIGN=** specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.
- **SAS 9.4M5** adds option **LABELFITPOLICY=STACKED**, which displays the Y or Y2 axis label vertically as stacked letters.

**DISCRETEOPTS=** supports the following new features for discrete axes:
- **TICKDISPLAYLIST=** specifies the text that is displayed for the tick values that are defined in the **TICKVALUELIST=** option.
- **TICKVALUEFITPOLICY=** supports new fit policies: **ROTATE**, **ROTALETEALWAYS**, **ROTALETEALWAYSDROP**, **SPLIT**, **SPLITALWAYS**, **SPLITALWAYSTHIN**, and **SPLITTHIN**.
- **TICKVALUELIST=** specifies the list of tick values that are displayed on the axis.
- **TICKVALUEROTATION=** specifies how the tick values are rotated on the X and X2 axes.
- **TICKVALUESPLITCHAR=** specifies a list of characters on which the tick values can be split if needed.
- **TICKVALUESPLITJUSTIFY=** specifies justification of the strings that are inside the tick value block.
- **TICKVALUESPLITCHARDROP=** specifies whether the split characters are included in the displayed tick values.
- **SAS 9.4M3** adds option **TICKVALUEFORMAT=**, which specifies how to format the values for major tick marks.
- **SAS 9.4M5** has the following enhancements:
  - **TICKVALUEROTATION=DIAGONAL2** rotates the tick values to a −45-degree diagonal position.
  - **TICKVALUEFITPOLICY=STACKEDALWAYS** always displays the tick values vertically as stacked letters.
  - **TICKVALUEFITPOLICY=STACKEDALWAYSTHIN** is the same as **STACKEDALWAYS**, except that thinning is performed when the tick values do not fit the available space.
LINEAROPTS= supports the following new features for linear axes:

- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
- MINORTICKS= specifies whether the minor tick marks are displayed on the axis.
- TICKDISPLAYLIST= specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.
- TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
- TICKVALUEFORMAT= now supports EXTRACTSCALETYPE=, which enables you to specify the type of scale that you want to extract.
- TICKVALUEROTATION= specifies how the tick values are rotated on the X and X2 axes.
- SAS 9.4M3 adds option TICKVALUEFORMAT=, which specifies how to format the values for major tick marks.
- SAS 9.4M5 adds option TICKVALUEROTATION=DIAGONAL2, which rotates the tick values to a –45-degree diagonal position.

LOGOPTS= supports the following new features for log axes:

- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
- TICKVALUELIST= specifies the tick values for a log axis as a space-separated list.
- TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
- TICKVALUEROTATION= specifies how the tick values are rotated on the X and X2 axes.
- SAS 9.4M3 adds option TICKVALUEFORMAT=, which specifies how to format the values for major tick marks.

TIMEOPTS= supports the following new features for time axes:

- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKINTERVAL= specifies the time interval between minor ticks.
- TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
- TICKVALUEROTATION= specifies how the tick values are rotated on the X and X2 axes.
SAS 9.4M3 has the following enhancements:

- **TICKVALUEFORMAT=** specifies how to format the values for major tick marks.
- **INTERVALMULTIPLIER=** specifies a multiplier to apply to the time interval that is in effect for the axis.

SAS 9.4M5 adds option **TICKVALUEROTATION=DIAGONAL2**, which rotates the tick values to a −45-degree diagonal position.

**LAYOUT DATALATTICE** and **LAYOUT DATAPANEL**:

- **LABELFITPOLICY=** specifies a policy for fitting axis labels in the available space.
- **LABELPOSITION=** specifies the position of the axis label.
- **LABELSPLITCHAR=** specifies one or more characters on which the axis labels can be split if needed.
- **LABELSPLITCHARDROP=** specifies whether the split characters are included in the displayed axis labels.
- **LABELSPLITJUSTIFY=** specifies the justification of the strings that are inside the axis label blocks.
- **TICKVALUEHALIGN=** specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.
- **TICKVALUEVALIGN=** specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.

SAS 9.4M5 adds option **LABELFITPOLICY=STACKED**, which displays the Y or Y2 axis label vertically as stacked letters.

**DISCRETEOPTS=** supports the following new features for discrete axes:

- **TICKDISPLAYLIST=** specifies the text that is displayed for the tick values that are defined in the **TICKVALUELIST=** option.
- **TICKVALUEFITPOLICY=** supports new fit policies: **ROTATE**, **ROTATEALWAYS**, **ROTATEALWAYSDROP**, **SPLIT**, **SPLITALWAYS**, **SPLITALWAYSTHIN**, and **SPLITTHIN**.
- **TICKVALUELIST=** specifies the list of tick values that are displayed on the axis.
- **TICKVALUEROTATION=** specifies how the tick values are rotated on the X and X2 axes.
- **TICKVALUESPLITCHAR=** specifies a list of characters on which the tick values can be split if needed.
- **TICKVALUESPLITJUSTIFY=** specifies justification of the strings that are inside the tick value block.
- **TICKVALUESPLITCHARDROP=** specifies whether the split characters are included in the displayed tick values.

SAS 9.4M3 adds option **TICKVALUEFORMAT=**, which specifies how to format the values for major tick marks.

SAS 9.4M5 has the following enhancements:

- **TICKVALUEROTATION=DIAGONAL2** rotates the tick values to a −45-degree diagonal position.
TICKVALUEFITPOLICY=STACKEDALWAYS always displays the tick values vertically as stacked letters.

TICKVALUEFITPOLICY=STACKEDALWAYSTHIN is the same as STACKEDALWAYS, except that thinning is performed when the tick values do not fit the available space.

LINEAROPTS= supports the following new features for linear axes:

- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
- MINORTICKS= specifies whether the minor tick marks are displayed on the axis.
- TICKDISPLAYLIST= specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.
- TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
- TICKVALUEFORMAT= now supports EXTRACTSCALETYPE=, which enables you to specify the type of scale that you want to extract.
- TICKVALUEROTATION= specifies how the tick values are rotated on the X and X2 axes.
- SAS 9.4M2 changes the MINORGRIDATTRS= option default to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.
- SAS 9.4M5 adds option TICKVALUEROTATION=DIAGONAL2, which rotates the tick values to a –45-degree diagonal position.

LOGOPTS= supports the following new features for log axes:

- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKCOUNT= specifies the number of minor ticks that are displayed on the axis.
- TICKVALUELIST= specifies the tick values for a log axis as a space-separated list.
- TICKVALUEPRIORITY= specifies whether the TICKVALUELIST= specification can extend the axis data range.
- VALUETYPES= specifies how the VIEWMIN=, VIEWMAX=, and TICKVALUELIST= option values are interpreted.
- SAS 9.4M2 changes the MINORGRIDATTRS= option default to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.
- SAS 9.4M3 adds option TICKVALUEFORMAT=, which specifies how to format the values for major tick marks.

TIMEOPTS= supports the following new features for time axes:
- MINORGRID= specifies whether grid lines are displayed at the minor tick values.
- MINORGRIDATTRS= specifies the attributes of the minor grid lines.
- MINORTICKINTERVAL= specifies the time interval between minor ticks.
- TICKVALUEFITPOLICY= now supports policies NONE and ROTATEALWAYS.
- TICKVALUEROTATION= specifies how the tick values are rotated on the X and X2 axes.

- **SAS 9.4M2** changes the MINORGRIDATTRS= option default to style element GraphMinorGridLines in order to visually contrast the major and minor grid lines.
- **SAS 9.4M3** adds option INTERVALMULTIPLIER=, specifies a multiplier to apply to the time interval that is in effect for the axis.
- **SAS 9.4M5** adds option TICKVALUEROTATION=DIAGONAL2, which rotates the tick values to a –45-degree diagonal position.

---

**Marker Definition Statement Enhancements**

**SYMBOLCHAR:**

**SAS 9.4M3** has the following enhancements:

- HOFFSET= and VOFFSET= now move the marker character within the marker character bounding box. The bounding box position remains centered on the data point.
- Offsets applied to the marker character with HOFFSET= and VOFFSET= are also applied to the marker character that is displayed in the legend.

---

**Discrete Attribute Map Enhancements**

The following enhancements have been made to discrete attribute maps:

**VALUE statement:**

- TEXTATTRS= specifies the text attributes to use when an attribute map is applied to text in a plot.

**SAS 9.4M1** has the following enhancements:

- You can now specify the attribute mapping information for a discrete attribute map in a SAS data set. You now have an alternative to coding your mapping information in a DISCRETEATTRMAP block in your template. For more information, see *SAS ODS Graphics: Procedures Guide*.
- DISCRETEATTRMAP statement:
  - DISCRETELEGENDENTRYPOLICY= specifies whether the items that the associated plot contributes to a discrete legend are items that appear only in the data or items that are defined only in the attribute map.
DISCRETEATTRVAR statement:

- ATTRMAP= now accepts a name that is specified in the ID column in a discrete attribute map data set. This feature enables you to create a discrete attribute map variable for a discrete attribute map that is defined in a SAS data set. To resolve the attribute map name in that case, you must specify the name of the attribute map data set in the DATTRMAP= option in the SGRENDER statement that renders the graph. For information about discrete attribute map data sets and the SGRENDER statement DATTRMAP= option, see SAS ODS Graphics: Procedures Guide.

VALUE statement:

- FILLATTRS= now supports the TRANSPARENCY= fill option.
- LINEATTRS= now supports the THICKNESS= line option.
- MARKERATTRS= now supports the SIZE=, TRANSPARENCY=, and WEIGHT= marker options.

Documentation Enhancements

In SAS 9.4M2, SAS Graph Template Language: User’s Guide is reorganized to make it easier to find information about how to use the Graph Template Language.
PART 1

Fundamentals

Chapter 1
Overview ................................................................. 3
Overview

Graph Template Language (GTL) .................................................... 3
GTL and the Output Delivery System (ODS) .................................. 3
A Quick Example ................................................................. 4
Template Compilation ............................................................ 5
Run-Time Actions ................................................................. 6
Components of a Graph ............................................................ 7
Graphical Layouts ................................................................. 8
Plots ...................................................................................... 10
Axes ...................................................................................... 11
Legends .................................................................................. 12
Flexible Templates ..................................................................... 13
Expressions and Functions ....................................................... 13
Dynamic Variables and Macro Variables .................................. 14
Conditional Logic ..................................................................... 15
Output ..................................................................................... 16
ODS GRAPHICS Statement ..................................................... 16
ODS Styles ............................................................................. 17
About the Examples in This Documentation ............................. 18
Examples and Resources on the Web ........................................ 19

Graph Template Language (GTL)

GTL and the Output Delivery System (ODS)

The SAS Graph Template Language (GTL) is an extension to the Output Delivery System (ODS) that enables you to create sophisticated graphics. For example, using GTL, you can generate Model-Fit plots, Distribution Plots, Comparative plots, Prediction Plots, and more.

The graphics produced by GTL are generated by template definitions that control the graph format and appearance and specify the variable roles to represent in the graph display. The graph can then be rendered by associating the templates with a data source.
The GTL templates are defined with PROC TEMPLATE. GTL includes conditional statements that can be used to determine what graph features are rendered. It also includes layout statements that specify the arrangement of graph features, plot statements that request specific plot types (such as histograms and scatter plots), and text and legend statements that specify titles, footnotes, legends, and other text-based graph elements.

The GTL templates are rendered using the SGRENDER procedure, which specifies a data source that contains appropriate data values and the template to use for rendering the graph.

You can also modify predefined GTL templates that SAS delivers for use on the SAS statistical procedures. For information about modifying existing templates, refer to SAS/STAT user’s guide.

This manual provides a complete reference to the Graph Template Language. For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

Note: If you are also a SAS/GRAPH user, then you might want to consult the SAS Graph Template Language: User’s Guide to learn about some of the distinctions between ODS Graphics and SAS/GRAPH.

A Quick Example

The data set Sashelp.Class is delivered with SAS. It includes data columns named Height and Weight, which store height and weight measures for a small sample of subjects. The Graph Template Language can be used to generate a histogram that shows the distribution of weight recorded in that data set:

The following SAS program produces the graph:
The DEFINE STATGRAPH statement on PROC TEMPLATE opens a definition block for defining a graphics template named HISTOGRAM. The HISTOGRAM template is stored in the template folder (also called the “template store,” by default located in Sasuser.Templat).

The template definition for HISTOGRAM specifies two GTL statements within a BEGINGRAPH/ENDGRAPH block: LAYOUT OVERLAY and HISTOGRAM.

The LAYOUT OVERLAY statement is one of the most fundamental layout statements. It can overlay the results of one or more plot statements, each of which shares the same plot area, axes, and legends. The layout in this example specifies only a single element: a HISTOGRAM with bars showing the distribution of observations of the data column named Weight.

The ENDLAYOUT statement ends the layout block, the ENDGRAPH statement ends the graph definition, and the END statement ends the template definition.

The ODS GRAPHICS statement uses the WIDTH= option to set a width for the output graph. Because the HEIGHT= option is not specified, GTL manages the graph’s aspect ratio and sets an appropriate height.

The DATA= option on PROC SGRENDER specifies Sashelp.Class as the data source for the graph. TEMPLATE= specifies HISTOGRAM as the template definition to use for rendering the graph.

Template Compilation

A GTL template describes the structure and appearance of a graph to be produced, similar to how a TABLE template describes the organization and content of a table.

All templates are stored, compiled programs. The following source program produces a simple GTL template named SCATTER:

```sas
proc template;
  define statgraph scatter;
  begingraph;
    layout overlay;
      scatterplot x=height y=weight;
    endlayout;
  endgraph;
end;
run;
```

The DATA= option on PROC SGRENDER specifies Sashelp.Class as the data source for the graph. TEMPLATE= specifies HISTOGRAM as the template definition to use for rendering the graph.
When this code is submitted, the statement keywords and options are parsed, just as with any other procedure. If no syntax error is detected, then an output template named SCATTER is created and stored in the default template folder Sasuser.Templat. No graph is produced. Note the following:

- Any required arguments in the template must be specified. In this example, X= and Y= in the SCATTERPLOT statement must specify variables for the analysis, but no checking for the existence of these variables is done at compile time. (Unlike other SAS procedures, PROC TEMPLATE does not perform a compile and then run sequence, which includes variable validation.)
- No reference to an input data set appears in the template.

Run-Time Actions

To produce a graph, a GTL template must be bound to a data source using the SGRENDER procedure. The following example uses SGRENDER to bind the SCATTER template to the SAS data set Sashelp.Class, which is delivered with SAS:

```sas
proc sgrender data=sashelp.class
template=scatter;
run;
```

Generally, an ODS data object is constructed by comparing the template references to column names with columns that exist in the current data set. In the current example, Sashelp.Class contains columns named Height and Weight. Because these column names match the columns that are named in template SCATTER, columns Height and Weight are added to the data object. The other columns in Sashelp.Class are ignored. (It is possible for a template to define new computed columns based on existing columns.)

After all the observations have been read, the data object and template definition are passed to a graph renderer, which produces the graph image. The image is then automatically integrated into the ODS destination. The visual properties of the graph are determined by the ODS style that is in effect.

Note: Template SCATTER is a restrictive definition: it can create a plot only with columns named Height and Weight. A GTL template can be made more flexible by
introducing dynamic variables or macro variables that supply variables and other information at run time. For more information, see “Flexible Templates” on page 13.

Components of a Graph

In general, a graph is made up of the following parts:

- titles and footnotes
- one or more cells that contain a composite of one or more plots
- legends, which can reside inside or outside a cell

The following figure shows the different parts of a graph:

**Figure 1.1 Components of a Graph**

1. **Graph**
   - a visual representation of data. The graph can contain titles, footnotes, legends, and one or more cells that have one or more plots.

2. **Cell**
   - a distinct rectangular subregion of a graph that can contain plots, text, and legends.
Graphical Layouts

One of the most powerful features of GTL is the syntax built around hierarchical statement blocks called “layouts.” The outermost layout block determines

- The overall organization of the graph—whether it uses a single-cell or a multi-cell display.
- What statements are allowed in the block. Generally, layout blocks can contain plots, lines of text, a legend, or even another layout.
- How the contained statements interact.

Table 1.1  Outermost Layouts in GTL

<table>
<thead>
<tr>
<th>Layout</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAY</td>
<td>General purpose layout for displaying 2-D plots in a single-cell.</td>
</tr>
<tr>
<td>OVERLAY3D</td>
<td>Layout for displaying 3-D plots in a single-cell.</td>
</tr>
<tr>
<td>OVERLAYEQUATED</td>
<td>Specialized OVERLAY with equated axes.</td>
</tr>
<tr>
<td>REGION</td>
<td>General purpose layout for displaying single-cell graphs that does not use axes.</td>
</tr>
<tr>
<td>GRIDDLED</td>
<td>Basic grid of plots. All cells are independent.</td>
</tr>
<tr>
<td>Layout</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LATTICE</td>
<td>Advanced multi-cell layout. Axes can be shared across columns or rows and be external to grid. Many grid labeling and alignment features.</td>
</tr>
<tr>
<td>DATALATTICE</td>
<td>Generates a classification panel from the values of 1 or 2 classifiers.</td>
</tr>
<tr>
<td>DATAPANEL</td>
<td>Generates a classification panel from the values of n classifiers.</td>
</tr>
<tr>
<td>GLOBALLEGEND</td>
<td>Specialized layout for creating a compound legend that contains multiple discrete legends.</td>
</tr>
</tbody>
</table>

For example, the following graph is a two-cell graph produced using the LAYOUT LATTICE statement as the outermost template in the layout.

The LAYOUT LATTICE statement is typically used to create a multi-cell layout of plots that are aligned across columns and rows. In the following template, which produced the graph, plot statements are specified within nested LAYOUT OVERLAY statements. Thus, the LATTICE automatically aligns the plot areas and tick display areas in the plots. The LATTICE layout is a good layout to choose when you want to compare the results of related plots.

```plaintext
proc template;
  define statgraph lattice;
  begingraph;
    entrytitle "Car Performance Profile";
    layout lattice / border=true pad=10 opaque=true
      rows=1 columns=2 columngutter=3;
```
Plots

The plots in GTL are classified in different ways, depending on the context of the discussion.

Within layout blocks, plots are often classified according to graphical dimension: whether they are projected in two or three visual dimensions. Thus, plots in GTL are often referred to as 2-D or 3-D plots, based on their graphical dimensions, not their data dimensions.

Relative to their input data, plots are classified according to the statements that calculate summary statistics from raw input data, and those that use calculated statistics as input parameters on the plot statement. Thus, many GTL plot statements have two versions: BARCHART and BARCHARTPARM, HISTOGRAM and HISTOGRAMPARM, and so on. The main distinction between such plots is the nature of the input data that they accept:

- The “non-parm” version (for example, BARCHART) computes its values from raw, unsummarized data. For example, a BARCHART computes the summary values it needs for the bars in the chart. Such plots are often referred to as “computed plots.”
- The “parm” version (for example, BARCHARTPARM) does not summarize or compute values from the input data but instead simply renders the input data it is given. Thus, the input data must be pre-summarized, perhaps by a SAS procedure. The “parm” version of plots, often referred to as “parameterized plots,” produce the same result as the non-parm version. However, they do not perform the calculations or data summarizations needed to achieve the result.

Chapter 5, “Key Concepts for Using Plots,” on page 189 discusses general concepts that apply across plot types. For detailed information about a particular plot, see the chapter for that plot.
GTL uses various criteria to determine the displayed axis features for a graph. Generally, axis features are based on the following criteria:

- the layout type
- the order of plot statements in the layout and the options specified on those statements
- the use of “primary” and “secondary” axes on the plots (when secondary axes are supported)
- the plot type
- the column(s) of data that contribute to defining the axis range
- the data formats for the contributing data columns

Depending on the layout type, 2-D plots can have up to four independent axes that can be displayed: X, Y, X2, and Y2. The X and Y axes are considered the primary axes, and the X2 and Y2 axes are considered the secondary axes. By default, the X2 and Y2 axes are not displayed. When requested, the secondary axes can be displayed as copies of the primary axes, or data can be mapped separately to them. The following figure identifies the X, Y, X2, and Y2 axes.

All 3-D plots display the standard X, Y, and Z axes.
Legends

Many plot statements support a GROUP= option that partitions the data into unique values, performs separate analysis, if necessary, and automatically assigns distinct visual properties to each group value. The visual properties of group values are defined by the style in effect.

Legends are not automatically displayed for plots with group values. Rather, an appropriate legend statement must be added to the template to generate the desired legend. In the following example, a legend is added to display markers and line patterns that show the association between the group values from a scatter plot and corresponding linear regression lines. The example shows the mechanism that GTL uses to associate a legend with its corresponding plot(s): a name is assigned to each plot that must be represented in the legend, and these names are then used as arguments for the legend statement (in this case, MERGEDLEGEND).

```plaintext
proc template;
  define statgraph scatterfit;
  begingraph;
  entrytitle "Linear Regression By Gender";
  layout overlay;
    scatterplot x=height y=weight / group=sex name="scat";
    regressionplot x=height y=weight/ group=sex name="reg";
    mergedlegend "scat" "reg" / border=true;
  endlayout;
  endgraph;
end;
run;
```
Flexible Templates

Several features in GTL can make template definitions less restrictive on input data and more general in nature. These features enable a single compiled template to produce many output variations.

Expressions and Functions

In GTL, expressions can be used to compute constants and data columns. The expressions must be enclosed in an EVAL construct. Within the expression, you can use DATA step functions, arithmetic operators, and other special functions supported by GTL.

Expressions are also useful in text statements like ENTRY and ENTRYTITLE. Both of these statements support rich text and have special text commands such as {SUP}, {SUB}, and {UNICODE}, which enable subscripting, superscripting, and Unicode characters.

The following template shows how the \( \pm \) symbol is included in the title line using its hexadecimal Unicode value. Also, new data columns are computed for the upper and lower error bars of the scatter plot, based on the input columns MeanWeight and STDERR.

```sas
proc template;
  define statgraph expression;
  begingraph;
  entrytitle "Errorbars show \( \pm \) \text{2 SE}";
  layout overlay;
  scatterplot x=age y=meanweight / yerrorlower=eval(meanweight - 2*stderr)
                        yerrorupper=eval(meanweight + 2*stderr);
  seriesplot x=age y=meanweight;
  endlayout;
end;
```
Dynamic Variables and Macro Variables

An extremely useful technique for generalizing templates is to define dynamic variables, macro variables, or both. The dynamic variables and macro variables resolve when the template is executed. The following PROC TEMPLATE statements can be used in a DEFINE STATGRAPH block:

<table>
<thead>
<tr>
<th>Template Statement</th>
<th>Purpose</th>
<th>Value supplied by...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNAMIC</td>
<td>defines one or more dynamic variables</td>
<td>either of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DYNAMIC= suboption of ODS= option of FILE PRINT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DYNAMIC statement of PROC SGRENDER</td>
</tr>
<tr>
<td>MVAR</td>
<td>defines one or more macro variables</td>
<td>%LET or CALL SYMPUT( )</td>
</tr>
<tr>
<td>NMVAR</td>
<td>defines one or more macro variables that resolve to a number or numbers</td>
<td>%LET or CALL SYMPUT( )</td>
</tr>
<tr>
<td>NOTES</td>
<td>provides information about the graph definition</td>
<td>user-supplied text</td>
</tr>
</tbody>
</table>

The following example defines a template named DYNAMICS that can create a histogram and density plot for any variable. It defines both macro variables and dynamic variables for run-time substitution. No data-dependent information is hard coded in the template.

Note: You can initialize macro variables with %LET statements and dynamic variables with SGRENDER’s DYNAMIC statement.

```plaintext
proc template;
define statgraph dynamics;
mvar SYSDATE9 SCALE;
nmvar BINS;
dynamic VAR VARLABEL;
begingraph;
    entrytitle "Histogram of " VAR;
    entrytitle "with Normal Distribution";
    layout overlay / xaxisopts={label=VARLABEL};
```
Conditional Logic

Using conditional logic, you can create templates that have multiple visual results or output representations, depending on existing conditions. The evaluation of a logical expression must generate one or more complete statements (not portions of statements). All conditional logic uses one of the following constructs:

<table>
<thead>
<tr>
<th>IF Block</th>
<th>IF/ELSE Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>if (condition)</td>
<td>if (condition)</td>
</tr>
<tr>
<td>statement(s);</td>
<td>statement(s);</td>
</tr>
<tr>
<td>endif;</td>
<td>else</td>
</tr>
<tr>
<td></td>
<td>statement(s);</td>
</tr>
<tr>
<td></td>
<td>endif;</td>
</tr>
</tbody>
</table>

In the IF statement, *condition* must be enclosed in parentheses. The condition can be any standard SAS expression involving arithmetic, logical operators, comparison operators, Boolean operators, or concatenation operators. The expression can also

For more information about using dynamic variables and macro variables, see Chapter 20, “Dynamic Variables and Macro Variables,” on page 1453.
use SAS DATA step functions. The expression resolves to a single numeric value, which is **true** or **false**.

In the following example, a histogram is conditionally overlaid with a normal distribution curve, a Kernel Density Estimate distribution curve, both, or neither:

```sas
proc template;
   define statgraph conditional;
   dynamic VAR VARLABEL BINS CURVE;
   begingraph;
      entrytitle "Histogram of " VAR;
      layout overlay / xaxisopts=(label=VARLABEL);
      histogram VAR / nbins=BINS;

      if (upcase(CURVE) in ("ALL" "KERNEL"))
         densityplot VAR / kernel() name="k"
            legendlabel="Kernel"
            lineattrs=(pattern=dash);
      endif;

      if (upcase(CURVE) in ("ALL" "NORMAL"))
         densityplot VAR / normal() name="n"
            legendlabel="Normal";
      endif;

      discretelegend "n" "k";
   endlayout;
   endgraph;
end;
run;
```

Note that the legend syntax does not have to be made conditional. At run time, each plot name in the legend is checked. If the plot does not exist, then its name is removed from the legend name list. If no names appear in the DISCRETELEGEND statement, then the legend "drops out" and the histogram size is adjusted to fill the remaining space.

For more information about using conditional logic, see Chapter 23, "Conditional Logic," on page 1473.

**Output**

When using GTL, you focus primarily on defining template definitions that produce specific graphs and generate a particular output layout. Ultimately, you must also customize the graphical environment to get the exact output that you desire. The ODS GRAPHICS statement is available for customizing the graphical environment, and ODS styles enable you to manage the output appearance.

**ODS GRAPHICS Statement**

The ODS GRAPHICS statement is used to modify the environment in which graphics templates are executed. The ODS GRAPHICS statement is used to control
- whether ODS Graphics is enabled
- the type and name of the image created
- the size of the image
- whether features such as scaling and anti-aliasing are used.

The following ODS GRAPHICS statement uses the HEIGHT= and WIDTH= options to set an aspect ratio for the output image.

```sas
ods graphics on / height=175px width=200px;
proc sgrender data=sashelp.class
   template=scatter;
run;
ods graphics off;
```

For more information about using the ODS GRAPHICS statement in GTL, see SAS Graph Template Language: User’s Guide. For a more complete discussion of the ODS GRAPHICS statement, see “ODS GRAPHICS Statement” on page 1483.

### ODS Styles

When any graphics template is executed, there is always an ODS style in effect that governs the appearance of the output. The following ODS statement sends graphics output to the RTF output destination using the LISTING style:

```sas
ods rtf style=listing;
ods graphics on / height=175px width=200px border=off;
proc sgrender data=sashelp.class
   template=scatter;
run;
ods graphics off;
ods rtf close;
```

Support for ODS styles is highly integrated into GTL syntax. By default, the graphical appearance features of most plot and text statements are mapped to corresponding style elements and associated attributes. Because of this, your output tables and graphs always have a reasonable overall appearance. Moreover, output for a given ODS destination has a consistent look (for example, table colors and graph colors do not clash).

The following figures show how a graph’s appearance can be changed by using references to style elements to set the graph’s appearance options. This technique permits changes in graph appearance by style modification instead of graphical template modification. The graphs in the figures are generated with the following GTL statement:

```sas
contourplotparm x=x y=y z=density /
   contourtype=fill nhint=9
   colormodel=ThreeColorRamp ;
```

The following style template shows the definition for the ThreeColorRamp style element:

```sas
style ThreeColorRamp /
   endcolor = GraphColors("gramp3cend")
   neutralcolor = GraphColors("gramp3cneutral")
```
startcolor = GraphColors("gramp3cstart");

For more information about the use of ODS styles in GTL, see SAS Graph Template Language: User’s Guide. For a more complete discussion of ODS styles, see SAS Output Delivery System: User’s Guide.

About the Examples in This Documentation

The example programs that are shown in this document often provide all of the code that you need to generate the graphs that are shown in the figures. We encourage you to copy and paste the example code into your SAS session and generate the graphs for yourself. The examples are written to be runnable in the SAS windowing environment and in SAS Studio. Unless otherwise noted, the examples use the default ODS destination. In the SAS windowing environment, the default ODS destination is ODS HTML. For information about results in SAS Studio, see SAS Studio: User’s Guide for your version of SAS Studio. You can find the documentation for all versions of SAS Studio on the SAS Studio documentation page on support.sas.com.

If you generate the example graphs using an HTML destination, they are typically rendered as 640 pixel by 480 pixel images using the HTMLBlue style. Because of size limitations, the graphs in this document are not shown in their default size. They
are scaled down to meet the size requirements of our documentation production system. When graphs that are produced with ODS graphics are reduced in size, several automatic processes take place to optimize the appearance of the output. Among the differences between default size graphs and smaller graphs are that the smaller graphs have scaled down font sizes. Also, their numeric axes might display a reduced number of ticks and tick values. Thus, the graphs that you generate from the example programs will not always look identical to the graphs that are shown in the figures. However, both graphs will accurately represent the data.

When you produce your own graphical output, you can change the graph size and attributes, if needed. The *SAS Graph Template Language: User’s Guide* explains how to set fonts, DPI, anti-aliasing, and other features that contribute to producing professional-looking graphics of any size in any output format.

---

### Examples and Resources on the Web

The SAS website contains a large number of examples that can help you visualize and code your graphs. The examples cover a range of SAS technologies including the ODS Graphics procedures.

- **Graphically Speaking** is a blog focused on using ODS Graphics for data visualization in SAS. The blog covers topics related to the ODS Graphics procedures, the SAS Graph Template Language, and the SAS ODS Graphics Designer.
  
  [http://blogs.sas.com/content/graphicallyspeaking/](http://blogs.sas.com/content/graphicallyspeaking/)

- **The SAS Training Post** is a blog that provides tutorials, tips, and practical information about SAS. Dr. Robert Allison frequently contributes to the blog. His posts use SAS/GRAPH and ODS Graphics for data visualization.

  [http://blogs.sas.com/content/sastraining/author/robertallison/](http://blogs.sas.com/content/sastraining/author/robertallison/)

- **The SAS Knowledge Base** contains an abundance of searchable samples and SAS Notes. You can browse by topic, search for a particular note or a particular technology such as the name of a procedure, and conduct other searches.


  **Note:** The SAS Knowledge Base content is currently available only in English.

- The Graphics Samples Output Gallery in the SAS Knowledge Base is a collection of graphs organized by SAS procedure. The graphs link to the source code. The gallery is maintained by SAS Technical Support.


- **The Focus Area Graphics** site provides a simple interface to business and analytical graphs. The site is maintained by the SAS Data Visualization team.


In addition, SAS offers instructor-led training and self-paced e-learning courses to help you get started with platform graphics software. For more information about the courses available, see sas.com/training.
Chapter 2

BEGINGRAPH Statement

........................................... 23
BEGINGRAPH Statement

Defines the outermost container for a graph template that is defined with \textit{GTL-statements}.

Requirements:
- All STATGRAPH template definitions must start with a BEGINGRAPH statement and end with an ENDGRAPH statement.
- The BEGINGRAPH block must contain one and only one layout block.
- The layout block and its nested layouts, if any, must contain at least one plot.

Syntax

\texttt{BEGINGRAPH }\langle\texttt{options}\rangle; \\
\quad \langle\texttt{GTL-global-statements}\rangle \\
\quad \texttt{GTL-layout-block} \\
\quad \langle\texttt{GTL-global-statements}\rangle \\
\texttt{ENDGRAPH};

Summary of Optional Arguments

\textbf{Appearance options}

- \texttt{ATTRPRIORITY=\texttt{AUTO} \mid \texttt{COLOR} \mid \texttt{NONE}}
  specifies a priority for cycling the group attributes.
- \texttt{BACKGROUND\texttt{COLOR}=	exttt{style-reference} \mid \texttt{color}}
  specifies the color of the graph background.
- \texttt{BORDER=\texttt{TRUE} \mid \texttt{FALSE}}
  specifies whether a border is drawn around the graph.
- \texttt{BORDER\texttt{ATTRS}=	exttt{style-element} \mid \texttt{style-element (line-options)} \mid \texttt{(line-options)}}
  specifies the attributes of the border line around the graph.
DATACOLORS=(color-list)
  specifies the list of fill colors that will replace the graph data colors from
  the GraphData1–GraphDataN style elements.

DATACONTRASTCOLORS=(color-list)
  specifies the list of contrast colors that will replace the graph data contrast
  colors from the GraphData1–GraphDataN style elements.

DATAFILLPATTERNS=(fill-pattern-list)
  specifies the list of fill patterns that will replace the graph data fill patterns
  from the GraphData1–GraphDataN style elements, if specified.

DATALINEPATTERNS=(line-pattern-list)
  specifies the list of line patterns that will replace the graph data line
  patterns from the GraphData1–GraphDataN style elements.

DATASKIN=None | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of all plots in the template that support
  data skins.

DATASYMBOLS=(marker-symbol-list)
  specifies the list of marker symbols that will replace the graph data marker
  symbols from the marker symbols that are defined in the GraphData1–
  GraphDataN style elements.

DESIGNHEIGHT=DEFAULTDESIGNHEIGHT | dimension
  specifies the design height of the graph.

DESIGNWIDTH=DEFAULTDESIGNWIDTH | dimension
  specifies the design width of the graph.

DRAWSPACE= GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
  specifies a global drawing space and drawing units for all of the draw
  statements within this BEGINGRAPH block.

OPAQUE=TRUE | FALSE
  specifies whether the graph background is opaque or transparent.

PAD=dimension | (pad-options)
  specifies the amount of extra space that is added inside the graph border.

SUBPIXEL=AUTO | OFF | ON
  specifies whether subpixel rendering is used for drawing smooth curved
  lines or for spacing bars more precisely.

Axis options

AXISBREAKSYMBOL=BRACKET | NOTCH | SLANTEDLEFT |
  SLANTEDRIGHT | SQUIGGLE | SPARK | Z
  specifies a symbol to use on the axis lines to indicate a break in the axis.

AXISBREAKTYPE=FULL | AXIS
  specifies whether the axis break is indicated in the full display or only on
  the axis line.

AXISLINEEXTENT=FULL | DATA | number
  specifies the extent of the axis line for all axes.

DISCRETEAXISOFFSETPAD=TRUE | FALSE
  specifies whether additional padding is added to the minimum and
  maximum axis offsets for discrete axes.

Label options

LABELPLACEMENT=AUTO | GREEDY | SA
specifies the label-placement algorithm to use for positioning labels in the graphs.

**SAPLACEMENTOPTS=(placement-options)**

specifies the options for the label-placement algorithm when **LABELPLACEMENT=SA**.

**Midpoint options**

**INCLUDEMISSINGDISCRETE=TRUE | FALSE**

specifies whether missing values are displayed on a discrete axis.

**Optional Arguments**

**ATTRPRIORITY=** **AUTO | COLOR | NONE**

specifies a priority for cycling the group attributes.

**AUTO**

honors the current state of the attribute priority rotation pattern as specified in the active style or in the ODS GRAPHICS statement.

**COLOR**

changes the current setting of attribute priority rotation pattern to the color-priority pattern by cycling through the list of colors while holding the marker symbol, line pattern, or fill pattern constant. When all of the colors are exhausted, the marker symbol, line style, or fill pattern attribute increment to the next element, and then the colors in the list are repeated. This pattern repeats as needed.

**NONE**

changes the current setting of attribute priority rotation pattern to the default pattern, which cycles progressively through the attribute lists.

**Tip**

Use **NONE** if you want groups to be distinguished by color, marker, and line changes for all ODS styles that use color.

**Default**

The attribute priority pattern that is specified in the active style or in the ODS GRAPHICS statement.

**Interactions**

This option overrides the attribute priority rotation pattern that is specified in the current style and the ATTRPRIORITY= option in the ODS GRAPHICS statement.

The default lists of data colors, contrast colors, marker symbols, and line patterns are set in the active style’s GraphData1–GraphDataN elements.

The individual attributes in these lists can be overridden with the BEGINGRAPH options **DATACOLORS=**, **DATACONTRASTCOLORS=**, **DATALINEPATTERNS=**, and **DATASYMBOLS=**.

The ATTRPRIORITY= option affects the cycling of the style attributes for **GROUP=**, **CYCLEATTRS=TRUE**, and explicit style references such as **MARKERATTRS=GraphData2**.

**See**

“Attribute Rotation Patterns” in **SAS Graph Template Language: User’s Guide**
AXISBREAKSYMBOL=BRACKET | NOTCH | SLANTEDLEFT | SLANTEDRIGHT | SQUIGGLE | SPARK | Z

specifies a symbol to use on the axis lines to indicate a break in the axis.

Note: This feature applies to SAS 9.4M3 and to later releases.

The following figure shows an example of each symbol on a horizontal linear axis for ranges 1–4 and 6–10.

<table>
<thead>
<tr>
<th>BRACKET</th>
<th>NOTCH</th>
<th>SLANTEDLEFT</th>
<th>SLANTEDRIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 6 8 10</td>
<td>1 3 6 8 10</td>
<td>1 3 6 8 10</td>
<td>1 3 6 8 10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SPARK</th>
<th>SQUIGGLE</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 3 6 8 10</td>
<td>1 3 6 8 10</td>
<td>1 3 5 8 10</td>
</tr>
</tbody>
</table>

Default: SQUIGGLE

Restriction: This option applies to linear and time axes only.

Requirements: The AXISBREAKTYPE= option must be set to AXIS for this option to have any effect.

You must use the INCLUDERANGES= option to specify ranges for the axis for this option to have any effect.

The DISPLAY= option for the axis must include the axis line for this option to have any effect.

AXISBREAKTYPE=FULL | AXIS

specifies whether the axis break is indicated in the full display or only on the axis line.

Note: This feature applies to SAS 9.4M3 and to later releases.

The following figure shows an example of each type for ranges 50–52 and 56–73 on a linear horizontal axis.

Figure 2.1 Axis Break Types FULL and AXIS

<table>
<thead>
<tr>
<th>FULL</th>
<th>AXIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>140 120 100 80 60 40 20 0</td>
<td>140 120 100 80 60 40 20 0</td>
</tr>
<tr>
<td>50 57 61 65 69 73</td>
<td>50 57 61 65 69 73</td>
</tr>
</tbody>
</table>

Default: FULL

Requirements: You must use the INCLUDERANGES= axis option to specify ranges for the axis for this option to have any effect.
The axis line or the plot wall outline must be displayed for AXIS to have any effect. Otherwise, FULL is used.

**Interaction**
When AXIS is specified, if the secondary axis line or the plot wall outline is displayed, then the axis break symbol is displayed on both the primary and the secondary axis. Otherwise, the break symbol is displayed only on the primary axis, as shown in Figure 2.2 on page 26.

**Notes**
The axis break indicators pass through inner margin areas.

No attempt is made to avoid collisions between the axis break indicators and other graphical elements.

**Tip**
When you use AXIS, use the AXISBREAKSYMBOL= option to change the break symbol.

**AXISLINEEXTENT=FULL | DATA | number**
specifies the extent of the axis line for all axes.

---

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**FULL**
specifies axis lines that extend along the entire length of the axis.

**DATA**
specifies axis lines that extend through the data range from the minimum offset to the maximum offset.

**number**
specifies, as a decimal proportion, how much the axis line extends from DATA toward FULL. A value of 0 is equivalent to DATA, and a value of 1 is equivalent to FULL.

**Range** 0–1

**Tip**
A numeric value is useful for bar charts when DATA terminates the axis line at the midpoint positions of the minimum and maximum bars. In that case, you can specify a numeric value to lengthen the axis line so that it extends to the full width of both bars.

The following figure shows a simple example of each value for the X axis and Y-axis lines. The light-blue dashed lines depict the minimum and maximum offsets that are set on the axes.

[Diagram showing examples of FULL, DATA, and 0.5 settings for X and Y axes]

**Default** FULL
Restriction: This option is valid only in OVERLAY and OVERLAYEQUATED layouts.

Tips: The graph wall outline might appear to be an axis line. In that case, use the WALLDISPLAY=NONE or WALLDISPLAY=(FILL) option in the layout statement to suppress the wall outline.

Use the LINEEXTENT= axis option to control the axis line extent on a per-axis basis.

**BACKGROUNDCOLOR=style-reference | color**

specifies the color of the graph background.

*style-reference*

specifies a style reference in the form `style-element:style-attribute`. Only the style attribute named COLOR or CONTRASTCOLOR is used.

**Default**
The GraphBackground:Color style reference.

**Interaction**
This option has no effect when `OPAQUE=FALSE`.

**BORDER=TRUE | FALSE**

specifies whether a border is drawn around the graph.

**Default**
The ODS GRAPHICS statement BORDER= option setting, which is TRUE by default.

**Interaction**
If this option is set to FALSE, then the BORDERATTRS= option is ignored.

**See**
“boolean ” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the border line around the graph.

**Default**
The GraphBorderLines style element.

**Interaction**
`BORDER= TRUE` must be set for this option to have any effect.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Line Options” on page 1522 for available `line-options`.

**DATACOLORS=(color-list)**

specifies the list of fill colors that will replace the graph data colors from the GraphData1–GraphDataN style elements.

*(color-list)*

a space-separated list of colors, enclosed in parentheses. You can use a style attribute reference such as GraphData3:color, a color name, or an RGB, CMYK, HLS, or HSV (HSB) color code to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement**
The list of colors must be enclosed in parentheses.

**Example**
`datacolors=(CXFF0000 green blue) `
When this option is specified, the fill colors rotate through this color list rather than through the colors that are defined in the GraphData1–GraphDataN style elements. For information about the attribute rotation patterns, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

**Default**

The colors that are defined in the GraphData1–GraphDataN style elements.

**Interaction**

Where applicable, the COLOR= suboption of the FILLATTRS= option overrides the DATACOLORS= option.

**DATACONTRASTCOLORS=**(color-list)

specifies the list of contrast colors that will replace the graph data contrast colors from the GraphData1–GraphDataN style elements.

*(color-list)*

a space-separated list of contrast colors, enclosed in parentheses. You can use a style attribute reference such as GraphData3:color, a color name, or an RGB, CMYK, HLS, or HSV (HSB) color code to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement**

The list of colors must be enclosed in parentheses.

**Example**

datacontrastcolors=(red graphdata3:contrastcolor CXFF0000)

When this option is specified, the contrast colors cycle through this color list rather than through the contrast colors that are defined in the GraphData1–GraphDataN style elements. For information about the attribute rotation patterns, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

**Default**

The contrast colors that are defined in the GraphData1–GraphDataN style elements.

**Interaction**

Where applicable, the COLOR= suboption of the MARKERATTRS= option or the LINEATTRS= option overrides the DATACONTRASTCOLORS= option.

**DATAFILLPATTERNS=**(fill-pattern-list)

specifies the list of fill patterns that will replace the graph data fill patterns from the GraphData1–GraphDataN style elements, if specified.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

*(fill-pattern-list)*

a space-separated list of fill patterns, enclosed in parentheses. You can use a style attribute reference such as GraphData3:fillpattern or a two-character fill-pattern name such as L1, X5, and so on, to specify a fill pattern. The list can contain a mix of style attribute references and pattern names.

**Requirement**

The list of fill patterns must be enclosed in parentheses.

**Example**

datafillpatterns=(graphdata4:fillpattern L5 X2)

When this option is specified, the fill patterns rotate through this fill-pattern list rather than through the fill patterns that are defined in the GraphData1–GraphDataN style elements, if any. For information about the attribute rotation...
patterns, see "Attribute Rotation Patterns" in SAS Graph Template Language: User's Guide.

Default The fill patterns that are defined in the GraphData1–GraphDataN style elements, if any.

Interaction Where applicable, the PATTERN= suboption of the FILLPATTERNATTRS= option overrides the DATAFILLPATTERNS= option.

DATALINEPATTERNS=(line-pattern-list)
specifies the list of line patterns that will replace the graph data line patterns from the GraphData1–GraphDataN style elements.

(line-pattern-list) a space-separated list of line patterns, enclosed in parentheses. You can use a style attribute reference such as GraphData3:lineStyle, a line pattern number, or a line pattern name (where applicable) to specify a pattern. The list can contain a mix of style attribute references, line pattern numbers, and line pattern names.

Requirement The list of line patterns must be enclosed in parentheses.

When this option is specified, the line patterns cycle through this line-pattern list rather than through the line patterns that are defined in the GraphData1–GraphDataN style elements. When the patterns in line-pattern-list are exhausted, the patterns repeat.

Default The line patterns that are defined in the GraphData1–GraphDataN style elements.

Interaction Where applicable, the PATTERN= suboption of the LINEATTRS= option overrides the DATALINEPATTERNS= option.

Example datalinepatterns=(solid dash 16 26)

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of all plots in the template that support data skins. The following plot statements support data skins:

BARCHART HISTOGRAM SCATTERPLOT
BARCHARTParm HISTOGRAMParm SCATTERPLOTMATRIX
BOXPLOT LINECHART SERIESPLOT
BOXPLOTParm NEEDLEPLOT STEPPLOT
BUBBLEPLOT PIECHART VECTORPLOT
DROPLINE POLYGONPLOT WATERFALLCHART
HIGHLOWPLOT REFERENCeline

Default The GraphSkins:DataSkin style attribute, if it is specified in the current style. If the current style does not specify the GraphSkins:DataSkin style attribute, then the default is NONE.

Restriction Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX=
option in your ODS GRAPHICS statement to increase the maximum limit.

Interaction
This option is overridden by the DATASKIN= option in the individual plot statements.

Note
Applying data skins to a graph that has a very large number of markers can negatively impact performance.

**DATASYMBOLS=**(marker-symbol-list)
specifies the list of marker symbols that will replace the graph data marker symbols from the marker symbols that are defined in the GraphData1–GraphDataN style elements.

(marker-symbol-list)
a space-separated list of marker symbols, enclosed in parentheses. You can use a style attribute reference such as GraphData5:markerSymbol or a marker symbol name to specify a marker. The list can contain a mix of style attribute references and marker symbol names.

Requirement
The list of **marker symbols** must be enclosed in parentheses.

When this option is specified, the marker symbols cycle through this marker symbol list rather than through the line patterns that are defined in the GraphData1–GraphDataN style elements. When the symbols in **marker-symbol-list** are exhausted, the symbols repeat.

Default
The marker symbols that are defined in the GraphData1–GraphDataN style elements.

Interaction
Where applicable, the SYMBOL= suboption of the MARKERATTRS= option overrides the DATASYMBOLS= option.

Example
datasyMBOLS= (circle square triangle star)

**DESIGNHEIGHT=DEFAULTDESIGNHEIGHT | dimension**
specifies the design height of the graph.

Default
DEFAULTDESIGNHEIGHT. This value is obtained from the SAS Registry key **ODS ➔ ODS Graphics ➔ Design Height** when the graph is rendered. The initial value of this registry key is 480px.

Restriction
The minimum **dimension** value that you can set is 2 pixels. If a smaller setting is specified, then the default design height is used.

Interactions
The design height can be overridden at run time with a render height that is specified with the HEIGHT= option in the ODS GRAPHICS statement (external to the template). Also, the ODS destination statement’s IMAGE_DPI= option can affect the height.

You can change the value of the Design Height registry key in the SAS registry. However, doing so affects the design height of all templates that do not include an explicit dimension for the design height. You can also change the height setting in the graph style, but doing so affects the height of all templates that use that style.

See
“**dimension**” on page 1480
DESIGNWIDTH=DEFAULTDESIGNWIDTH | dimension
specifies the design width of the graph.

Default

DEFAULTDESIGNWIDTH. This value is obtained from the SAS Registry key ODS ➔ ODS Graphics ➔ Design Width when the graph is rendered. The initial value of this registry key is 640px.

Restriction

The minimum dimension value that you can set is 2 pixels. If a smaller setting is specified, then the default design width is used.

Interactions

The design width can be overridden at run time with a render width that is specified with the WIDTH= option in the ODS GRAPHICS statement (external to the template). Also, the ODS destination statement’s IMAGE_DPI= option can affect the width.

You can change the value of the Design Width registry key in the SAS registry. However, doing so affects the design width of all templates that do not include an explicit dimension for the design width. You can also change the width setting in the graph style, but doing so affects the width of all templates that use that style.

See

"dimension" on page 1480

DISCRETEAXISOFFSETPAD=TRUE | FALSE
specifies whether additional padding is added to the minimum and maximum axis offsets for discrete axes. When set to TRUE, an additional 5 pixels of padding are added to the minimum and maximum axis offsets.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default

TRUE

Restriction

This option applies only to discrete axes.

Tip

This option is useful for heat maps when you want the heat map to occupy the entire plot area. In that case, in addition to setting this option to FALSE, set OFFSETMIN= and OFFSETMAX= to 0 for the discrete axes.

DRAWSPACE= GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies a global drawing space and drawing units for all of the draw statements within this BEGINGRAPH block.

Default

LAYOUTPERCENT

Tip

Individual draw statements within this BEGINGRAPH block can override this global setting.

See

"About the Drawing Space and Drawing Units" on page 1322

INCLUDEMISSINGDISCRETE=TRUE | FALSE
specifies whether missing values are displayed on a discrete axis.

Default

FALSE
Interaction This option affects all charts and plots within the template.

See “boolean” on page 1479 for other Boolean values that you can use.

**LABELPLACEMENT=** AUTO | GREEDY | SA

specifies the label-placement algorithm to use for positioning labels in the graphs. The following labels are affected:

- data labels for needle plots, scatter plots, series plots, step plots, and vector plots
- vertex labels for line charts
- curve labels when the curve label is positioned at the start or end of the curve

**AUTO**
always selects GREEDY.

**GREEDY**
specifies the Greedy method for managing label collision. The Greedy method tries different placement combinations in order to find an optimal approximation that avoids collisions. Label placement using this method is often less optimal than label placement using the Simulated Annealing (SA) method. However, depending on the number of data points and the potential for label collisions, the Greedy process can be significantly faster.

**SA**
specifies the Simulated Annealing method for managing label collision. The SA method attempts to determine the global minimization-of-cost function, which is based on a simulated annealing algorithm. The resulting label placement is usually better than placement using the Greedy method. However, depending on the number of data points and the potential for label collisions, the SA method can be significantly slower.

**Restriction**
For BANDPLOT and LINECHART, the SA method has no effect on the curve labels when the CURVELABELPOSITION= option specifies START or END.

**Default**
The value specified by the ODS GRAPHICS statement LABELPLACEMENT= option, which is AUTO by default.

**Restriction**
The data label placement algorithm is not aware of bar labels, curve labels, box plot outlier labels, and marker characters. Collisions between these elements and data labels might occur regardless of the LABELPLACEMENT= setting.

**Interactions**
This option overrides the ODS GRAPHICS statement LABELPLACEMENT= option.

This option affects a plot's labels only when DATALABELPOSITION=AUTO is in effect for that plot.

The data label font size might be reduced in order to avoid overlapping labels and markers. Starting with SAS 9.4M3, when a broken axis is used, the data-label font size is not scaled during label placement.

**OPAQUE=** TRUE | FALSE
specifies whether the graph background is opaque or transparent.
Note: This feature applies to SAS 9.4M3 and to later releases.

When this option is set to FALSE, the graph background is transparent.

Default TRUE

Restriction A transparent background is supported only by the PNG, EMF, PDF, and SVG output formats. The PS output format supports a transparent background when the graph is rendered as a PNG image. It does not support a transparent background when the graph is rendered as vector-graphics output.

Interaction When this option is set to FALSE, the BACKGROUNDCOLOR= option has no effect.

PAD=dimension | (pad-options)

specifies the amount of extra space that is added inside the graph border.

dimension

specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(pad-options)

a space-separated list of one or more of the following name-value-pair options enclosed in parentheses:

LEFT=dimension specifies the amount of extra space added to the left side.

RIGHT=dimension specifies the amount of extra space added to the right side.

TOP=dimension specifies the amount of extra space added to the top.

BOTTOM=dimension specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use pad-options to create non-uniform padding.

Default Padding for all sides is 10 pixels.

Note The default units for dimension are pixels.

See “dimension” on page 1480

SAPLACEMENTOPTS=(placement-options)

specifies the options for the label-placement algorithm when LABELPLACEMENT=SA. Placement options can be any of the following:

MAXITERATIONS=positive-integer specifies the maximum number of iterations for the SA label-placement algorithm.

Default 100
WEIGHTS=(keyword-number-list)
specifies the relative weight to give to a particular cost when determining the best label position. The keyword number list is a space-separated list of keyword = number pairs.

The following keywords can be used:

- **LABEL**: assigns a weight to the overlapping of labels
- **MARKER**: assigns a weight to the overlapping of markers and labels
- **OUTOFBOUND**: assigns a weight to labels that are out-of-bounds or clipped
- **PRIORITY**: assigns a weight to the priority of each potential label position
- **OBSTACLE**: assigns a weight to the overlapping of labels with drop lines, needles, reference lines, series lines, step lines, and vector lines

The higher the number, the more weight is assigned to the specified cost. For example, if MARKER is given more weight than OBSTACLE, avoiding marker collisions is given a higher priority than avoiding line collisions.

Default: A weight of 1.0 is assigned to each keyword

Restriction: A dynamic variable reference or a run-time macro variable reference cannot be used to specify a keyword value.

Example:
```plaintext
saplacementopts=(maxiterations=100
               weights=(LABEL=2.0 OBSTACLE=10.0))
```

**SEED=positive-integer**
specifies a random number seed for the Simulated Annealing algorithm.

Default: 1234567

Range: 0–2147483647 (2^{31}–1), where 0 specifies the current Java time as the seed value

Restriction: This option applies only when LABELPLACEMENT=SA.

**SUBPIXEL=AUTO | OFF | ON**
specifies whether subpixel rendering is used for drawing smooth curved lines or for spacing bars more precisely.

Note: Starting with **SAS 9.4M3**, this option controls subpixel rendering only for image output. For vector-graphics output, subpixel rendering is always enabled.

**AUTO**
the system determines whether to use subpixel rendering. In **SAS 9.4M2** and in earlier releases, the system uses the default rendering for the rendering technology. Starting with **SAS 9.4M3**, for image output, if the SUBPIXEL= option is explicitly set in an ODS GRAPHICS statement, the system honors its setting. Otherwise, the system determines whether to use subpixel rendering based on the following criteria:
If a SCATTERPLOT or SCATTERPLOTMATRIX statement is used, subpixel rendering is OFF for the graph.

If neither a SCATTERPLOT nor a SCATTERPLOTMATRIX statement is used, subpixel rendering is turned ON for the graph if one or more of the following statements is also used:

- BANDPLOT
- DENSITYPLOT
- LOESSPLOT
- BARCHART
- HEATMAP
- PBSPLINEPLOT
- BARCHARTPARM
- HEATMAPPARM
- POLYGONPLOT
- BOXPLOT
- HIGHLOWPLOT
- REGRESSIONPLOT
- BOXPLOTPARM
- HISTOGRAM
- SERIESPLOT
- BUBBLEPLOT
- HISTOGRAMPARM
- WATERFALLCHART
- CONTOURPLOTPARM
- LINECHART

For all other cases, subpixel rendering is turned OFF.

OFF
never uses subpixel rendering.

Note OFF is valid starting with SAS 9.4M3.

ON
always uses subpixel rendering, when applicable, for image output when rendering graphs.

Default AUTO

Restrictions
In SAS 9.4M2 and in earlier releases, subpixel rendering can be used only for the following statements: BANDPLOT, BARCHART, BARCHARTPARM, DENSITYPLOT, LINECHART, LOESSPLOT, PBSPLINEPLOT, REGRESSIONPLOT, and SERIESPLOT. Starting with SAS 9.4M3, subpixel rendering can be used for all plots and charts.

Starting with SAS 9.4M3, this option is ignored for vector-graphics output.

Requirement Anti-aliasing must be enabled for this option to have any effect.

Interaction Starting with SAS 9.4M3, this option overrides the SUBPIXEL= option in the ODS GRAPHICS statement.

Tips If anti-aliasing is disabled, use the ANTIALIAS=ON option in the ODS GRAPHICS statement to enable it.

Anti-aliasing is disabled automatically when the resources required for anti-aliasing exceed a preset threshold. When anti-aliasing is disabled for all or part of a graph, subpixel rendering is disabled for the entire graph. A note is written to the SAS log that provides information about how to use the ANTIALIASMAX= option in an ODS GRAPHICS statement to re-enable anti-aliasing.

See “Using Subpixel Rendering” in SAS Graph Template Language: User’s Guide

“ODS GRAPHICS Statement” on page 1483 for information about the ANTIALIAS= and ANTIALIASMAX= options.
Details

About the BEGINGRAPH Statement

All template definitions in the Graphics Template Language must start with a BEGINGRAPH statement and end with an ENDGRAPH statement. Within a BEGINGRAPH block, one and only one GTL layout block is required. It can be a LATTICE, GRIDDED, OVERLAY, OVERLAYEQUATED, OVERLAY3D, REGION, DATALATTICE, or DATAPANEL layout block. This layout block and its nested layouts, if any, must contain at least one plot statement. It can contain other nested layout blocks.

The GTL global statements apply to the entire template and can include ENTRYTITLE and ENTRYFOOTNOTE statements, attribute maps, draw statements, conditional statements, and so on. Any of these global statements can precede or follow the GTL layout block.

Changing the Size of Your Graph

By default, graphs are rendered at 640px by 480px (4:3 aspect ratio). To change the output size for a single graph, use the DESIGNWIDTH= and DESIGNHEIGHT= options in the BEGINGRAPH statement for that graph. For example, the template in the "Example Program" on page 38 uses DESIGNHEIGHT= to change the graph height to 320px. To prevent the graph width from automatically scaling to preserve the 4:3 aspect ratio, it uses DESIGNWIDTH= to maintain the 640px width. In this instance, the setting renders each graph cell as a 320px by 320px square. (The cells are square in this case, but the resulting cell size depends on the graph definition and would not be the same for all graphs.)

Note: To change the graph sizes for all templates in the current SAS session, you can use the WIDTH= and HEIGHT= options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement and remain in effect unless they are changed in another ODS GRAPHICS statement. You can also use WIDTH= and HEIGHT= settings in the graph style to modify the graphs sizes across template definitions. Be aware, however, that if you explicitly manage the graph output size, then the graph elements might be scaled so that the size specification is honored.

The following template defines a square graph (equal height and width, 1:1 aspect ratio) by setting the design width equal to the internal default height (480px). The setting is made with DESIGNWIDTH=DEFAULTDESIGNHEIGHT:

Note: A "square graph" means that the output graph’s width and height are equal. That does not imply that the X axis and Y-axis lengths are equal if the graph contains only one cell.

```plaintext
proc template;
define statgraph squareplot;
dynamic title xvar yvar;
begingraph / designwidth=defaultDesignHeight;
  entrytitle title;
  layout overlayequated / equatetype=square;
    scatterplot x=xvar y=yvar;
    regressionplot x=xvar y=yvar;
  endlayout;
endgraph;
```
If this template is executed with the following SGRENDER procedure statement, then a 480px by 480px graph is created:

```plaintext
proc sgrender data=mydata template="squareplot";
dynamic title="Square Plot" xvar="time1" yvar="time2";
run;
```

If the ODS GRAPHICS statement's WIDTH= or HEIGHT= options change the render width or render height, then the *squareplot* template's 1:1 aspect ratio would still be honored. Thus, both of the following GRENDER procedure statements would create a 550px by 550px graph:

```plaintext
ods graphics / width=550px;
proc sgrender data=mydata template="squareplot";
dynamic title="Square Plot" xvar="time1" yvar="time2";
run;
```

```plaintext
ods graphics / height=550px;
proc sgrender data=mydata template="squareplot";
dynamic title="Square Plot" xvar="time1" yvar="time2";
run;
```

**Example: BEGINGRAPH Statement**

The following graph was generated by the "Example Program" on page 38:

![Graph](image)

**Example Program**

The BEGINGRAPH statement block is a required outermost container for any graph template. One of its purposes is to support options that apply to the entire graph. For example, the default graph size that a template produces is typically 640x480 pixels. If you need a different size, then you can declare the alternative size on this statement. To do so, use the DESIGNWIDTH= option, or the DESIGNHEIGHT= option, or both. This program shows one way to set the width and height of two graph cells to be equal.

```plaintext
proc template;
```
define statgraph begingraph;
  dynamic XVAR YVAR;
  begingraph / designwidth=640px designheight=320px;
  layout lattice / columns=2;
    layout overlayequated / equatetype=square;
      entry "Linear Regression Fit" /
        valign=top textattrs=(weight=bold);
      scatterplot x=XVAR y=YVAR / datatransparency=0.5;
      regressionplot x=XVAR y=YVAR;
    endlayout;
    layout overlayequated / equatetype=square;
      entry "Loess Fit" /
        valign=top textattrs=(weight=bold);
      scatterplot x=XVAR y=YVAR / datatransparency=0.5;
      loessplot x=XVAR y=YVAR;
    endlayout;
  endlayout;
endgraph;
run;

proc sgrender data=sashelp.iris template=begingraph;
  dynamic title="Square Plot"
    xvar="SepalLength" yvar="SepalWidth";
run;
Summary of Layout Statements

Overview

Layout blocks always begin with the LAYOUT keyword followed by a keyword indicating the purpose of the layout. All layout blocks end with an ENDLAYOUT statement.

The following sections summarize the available layouts. To learn more about a layout, see the chapter devoted to that layout.
### Single-cell Layouts

<table>
<thead>
<tr>
<th>Layout (Description)</th>
<th>Graphics Allowed and Cells Produced</th>
<th>Comments</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAY (Single Cell)</td>
<td>2-D (1 cell)</td>
<td>General purpose layout for superimposing 2-D plots</td>
<td><img src="image1.png" alt="Example" /></td>
</tr>
<tr>
<td>OVERLAYEQUATE D (Single Cell)</td>
<td>2-D (1 cell)</td>
<td>Specialized OVERLAY with equated axes</td>
<td></td>
</tr>
<tr>
<td>PROTOTYPE (Single Cell)</td>
<td>2-D (1 cell)</td>
<td>Specialized LAYOUT used only as child layout of DATAPANEL or DATALATTICE</td>
<td></td>
</tr>
<tr>
<td>REGION (Single Cell)</td>
<td>2-D (1 cell)</td>
<td>General purpose layout for displaying a single-cell plot that does not use axes</td>
<td></td>
</tr>
<tr>
<td>OVERLAY3D (Single Cell)</td>
<td>3-D (1 cell)</td>
<td>General purpose 3-D layout for superimposing 3-D plots.</td>
<td><img src="image2.png" alt="Example" /></td>
</tr>
</tbody>
</table>
## Multi-cell Layouts

<table>
<thead>
<tr>
<th>Layout (Description)</th>
<th>Graphics Allowed and Cells Produced</th>
<th>Comments</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>LATTICE (Advanced Multi-cell)</td>
<td>2-D (1 or more cells)</td>
<td>All cells must be predefined. Axes can be shared across columns or rows and be external to grid. Many grid labeling and alignment features.</td>
<td><img src="example1.png" alt="Example" /></td>
</tr>
<tr>
<td>GRIDDED (Simple Multi-cell)</td>
<td>2-D (1 or more cells)</td>
<td>All cells must be predefined. Axes independent for each cell. Very simple multi-cell container.</td>
<td><img src="example2.png" alt="Example" /></td>
</tr>
</tbody>
</table>
# Data-driven Layouts

<table>
<thead>
<tr>
<th>Layout (Description)</th>
<th>Graphics Allowed and Cells Produced</th>
<th>Comments</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAPANEL (Classification Panel)</td>
<td>2-D (1 or more cells)</td>
<td>Displays a panel of similar graphs based on data subsets by classification variable(s). Number of cells is based on crossings of ( n ) classification variable(s).</td>
<td></td>
</tr>
</tbody>
</table>

| DATALATTICE (Classification Panel) | 2-D (1 or more cells) | Displays a panel of similar graphs based on data subsets by classification variable(s). Number of cells is based on crossings of 1 or 2 classification variables. |

## Legend Layout

<table>
<thead>
<tr>
<th>Layout</th>
<th>Cells Produced</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBALLEGEND</td>
<td>1 cell for a legend</td>
<td>Specialized layout for creating a compound legend that contains multiple discrete legends.</td>
</tr>
</tbody>
</table>
LAYOUT DATALATTICE Statement

Creates a grid of graphs based on one or two classification variables and a graphical prototype. By default, a separate instance of the prototype (a data cell) is created for each possible combination of the classification variables.

Restriction: You can specify only one LAYOUT PROTOTYPE block in the LAYOUT DATALATTICE block. If you specify more than one, then only the last prototype block specified is honored. The remaining prototype blocks are ignored.

Requirement: You must specify at least one ROWVAR= option or one COLUMNVAR= option. You can specify both.

Syntax

```
LAYOUT DATALATTICE ROWVAR=class-variable
COLUMNVAR=class-variable <options>;
  LAYOUT PROTOTYPE <options>;
    GTL-statements;
  ENDLAYOUT;
<SIDEBAR <options>;
```
GTL-statements;
ENDSIDEBAR;
ENDLAYOUT;
LAYOUT DATALATTICE COLUMNVAR=class-variable <options>;
    layout-prototype-block ;
    <sidebar-block(s)> ;
ENDLAYOUT;
LAYOUT DATALATTICE ROWVAR=class-variable <options>;
    layout-prototype-block ;
    <sidebar-block(s)> ;
ENDLAYOUT;

Summary of Optional Arguments

Appearance options

`BACKGROUNDCOLOR=style-reference | color`
specifies the color of the layout background.

`BORDER=TRUE | FALSE`
specifies whether a border is drawn around the layout.

`BORDERATTRS=style-element | style-element (line-options) | (line-options)`
specifies the attributes of the border line around the layout.

`CELLHEIGHTMIN=dimension`
specifies the minimum height of a cell in the grid.

`CELLWIDTHMIN=dimension`
specifies the minimum width of a cell in the grid.

`COLUMNGUTTER=dimension`
specifies the amount of empty space that is between the columns.

`COLUMNHEADERS=TOP | BOTTOM | BOTH`
specifies where to position the outside column heading.

`HEADERBACKGROUNDCOLOR=style-reference | color`
specifies the background color of the cell headers.

`HEADERBORDER=TRUE | FALSE`
specifies whether a border is drawn around the header cells.

`HEADERLABELATTRS=style-element | style-element (text-options) | (text-options)`
specifies the color and font attributes of the data labels.

`HEADERLABELLOCATION=OUTSIDE | INSIDE`
indicates whether the cell header is placed within each cell (INSIDE) or as row and column headers external to the lattice (OUTSIDE).

`HEADEROPAQUE=TRUE | FALSE`
specifies whether the background for cell headers is opaque (TRUE) or transparent (FALSE).

`HEADERPACK=TRUE | FALSE`
specifies whether the header cells are consolidated into a comma-separated list in order to save space.

`HEADERSEPARATOR="string"`
specifies one or more characters to place between each value in the cell header when HEADERPACK=TRUE.

`HEADERSPLITCOUNT=positive-integer`
specifies the number of headers to consolidate on a header line before splitting the text to the next line.

**OPAQUE**=TRUE | FALSE
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

**OUTERPAD**=AUTO | dimension | (pad-options)
specifies the amount of extra space to add outside the layout border.

**PAD**=dimension | (pad-options)
specifies the amount of extra space that is added inside the layout border.

**SHRINKFONTS**=TRUE | FALSE
specifies whether fonts in the layout are scaled down.

**SORTORDER**=(COLUMNVAR=sort-order-keyword | ROWVAR=sort-order-keyword)
specifies the order of the cells along the columns and rows.

**Axis options**

**COLUMN2AXISOPTS**=(axis-options)
specifies X2-axis options for all columns.

**COLUMN2DATARANGE**=AUTO | UNIONALL | UNION
specifies how the X2-axes of instances of the graph-prototype are scaled.

**COLUMNAXISOPTS**=(axis-options)
specifies X-axis options for all columns.

**COLUMNDATARANGE**=AUTO | UNIONALL | UNION
specifies how the X-axes of instances of the graph-prototype are scaled.

**ROW2AXISOPTS**=(axis-options)
specifies Y2-axis options for all rows.

**ROW2DATARANGE**=AUTO | UNIONALL | UNION
specifies how the Y2-axes of instances of the graph-prototype are scaled.

**ROWAXISOPTS**=(axis-options)
specifies Y-axis options for all rows.

**ROWDATARANGE**=AUTO | UNIONALL | UNION
specifies how the Y-axes of instances of the graph-prototype are scaled.

**Inset options**

**INSET**=(variable-list)
specifies what information is displayed in an inset.

**INSETOPTS**=(appearance-options)
specifies location and appearance options for the inset information.

**Layout options**

**COLUMNS**=integer
specifies the number of columns in the layout.

**COLUMNWEIGHT**=EQUAL | PROPORTIONAL
specifies how weights are assigned to the columns widths.

**HEADERLABELDISPLAY**=NAMEVALUE | VALUE | NONE
specifies the content of the cell headers.

**INCLUDEMISSINGCLASS**=TRUE | FALSE
specifies whether to include grid cells for crossings of the ROWVAR and COLUMNVAR variables that contain a missing value.

**PANELNUMBER**=positive-integer
specifies the number of the panel to produce.
ROWGUTTER=dimension
    specifies the amount of empty space between the rows.
ROWHEADERS=RIGHT | LEFT | BOTH
    specifies where to position the outside row heading.
ROWS=integer
    specifies the number of rows in the layout.
ROWWEIGHT=EQUAL | PROPORTIONAL
    specifies how weights are assigned to the row heights.
SKIPEMPTYCELLS=TRUE | FALSE
    specifies whether the external axes skip the empty cells in a partially filled grid.
START=TOPLEFT | BOTTOMLEFT
    indicates whether to start populating the grid from the top left or bottom left corner.

Required Arguments
You must specify at least one of the following arguments. You can specify both.

ROWVAR=class-variable
    specifies the classification variable for the rows. One row of cells is created for each unique value of the row class variable.
    See ROWS= option and “Managing Rows and Columns” on page 70

COLUMNVAR=class-variable
    specifies the classification variable for the columns. One column is created of each unique value of the column class variable.
    See COLUMNS= option and “Managing Rows and Columns” on page 70

Optional Arguments

BACKGROUNDCOLOR=style-reference | color
    specifies the color of the layout background.
    style-reference
        specifies a style reference in the form style-element:style-attribute.
        Only the style attribute named COLOR or CONTRASTCOLOR is used.
    color
        specifies a color by name.
    See color on page 1480

Default
    The GraphBackground:Color style reference.

Interaction
    OPAQUE=TRUE must be in effect in order for the color to be seen. By default, OPAQUE=FALSE.

BORDER=TRUE | FALSE
    specifies whether a border is drawn around the layout.

Default
    FALSE
Interaction If this option is set to FALSE, then the BORDERATTRS= option is ignored.

See “boolean” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=** *style-element | style-element (line-options) | (line-options)*
specifies the attributes of the border line around the layout.

**Default** The GraphBorderLines style element.

Interaction **BORDER= TRUE** must be set for this option to have any effect.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a **style-element**.

“Line Options” on page 1522 for available **line-options**.

**CELLHEIGHTMIN=** *dimension*
specifies the minimum height of a cell in the grid. Use this option in conjunction with the **CELLWIDTHMIN=** option to set the minimum cell size.

The overall size of the panel is constrained by the **HEIGHT=** and **WIDTH=** options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum height threshold for all cells. If the actual cell height becomes smaller, then no panel is drawn.

**Default** 100px

See “dimension” on page 1480

**CELLWIDTHMIN=** *dimension*
specifies the minimum width of a cell in the grid. Use this option in conjunction with the **CELLHEIGHTMIN=** option to set the minimum cell size.

The overall size of the panel is constrained by the **HEIGHT=** and **WIDTH=** options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum width threshold for all cells. If the actual cell width becomes smaller, then no panel is drawn.

**Default** 100px

See “dimension” on page 1480

**COLUMNAXISOPTS=** *(axis-options)*
specifies X-axis options for all columns.

**Restriction** Axis options must be enclosed in parentheses and separated by spaces.

See “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” for a list of options.

**COLUMN2AXISOPTS=** *(axis-options)*
specifies X2-axis options for all columns.
Restriction

This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement’s XAXIS= option to map data to the X2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974.

Requirement

Axis options must be enclosed in parentheses and separated by spaces.

See

“Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” for a list of options.

COLUMNDATARANGE=AUTO | UNIONALL | UNION

specifies how the X-axes of instances of the graph-prototype are scaled.

AUTO

selects the X-axis scale based on the COLUMNWEIGHT= option.

The X-axis scale is determined, as follows:

- When COLUMNWEIGHT=EQUAL (default), UNIONALL is selected.
- When COLUMNWEIGHT=PROPORTIONAL and the column axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

UNIONALL

scales the X-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION

scales the X-axis data ranges separately for each column on a per-panel basis.

The scaling does not span multiple panels.

Default: AUTO

Tip

Use the COLUMNAXISOPTS= option to control shared axis features.

See

The COLUMNWEIGHT= option.

The PANELNUMBER= option for information about how to create multiple panels.

COLUMN2DATARANGE=AUTO | UNIONALL | UNION

specifies how the X2-axes of instances of the graph-prototype are scaled.

AUTO

automatically selects the X2-axis scale based on the COLUMNWEIGHT= option and the column axis type.

The X2-axis scale is selected as follows:

- When COLUMNWEIGHT=EQUAL (default), UNIONALL is selected.
- When COLUMNWEIGHT=PROPORTIONAL and the column axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

UNIONALL

scales the X2-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).
UNION scales the X2-axis data ranges separately for each column on a per-panel basis. The scaling does not span multiple panels.

Default: AUTO

Interaction: This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement’s XAXIS= option to map data to the X2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974.

Tip: Use the COLUMN2AXISOPTS= option to control shared axis features.

See: The COLUMNWEIGHT= option.

The PANELNUMBER= option for information about how to create multiple panels.

COLUMNNGUTTER=\textit{dimension}
specifies the amount of empty space that is between the columns.

Default: 0

Note: If there are $n$ columns, then there are $n-1$ gutters.

See: “\textit{dimension}” on page 1480

COLUMNHEADERS=TOP | BOTTOM | BOTH
specifies where to position the outside column heading.

TOP
specifies that column heading text appears at the top of the layout.

BOTTOM
specifies that column heading text appears at the bottom of the layout.

BOTH
specifies that column heading text alternates between the top and bottom of the layout column by column.

Default: TOP

Interaction: HEADERLABELLOCATION= OUTSIDE must be set for this option to have any effect.

COLUMNS=\textit{integer}
specifies the number of columns in the layout.

Defaults
If this option is not specified, then the number of columns is dynamically adjusted to equal the number of classifier values for the COLUMVAR= variable.

If this option is specified, that many columns are created. If the number of COLUMNVAR classifier values is greater than the specified number of columns, then no graph is created for some classifier values. If the number of classifier values is smaller than
the specified number of columns, then extra empty columns are created.

Interactions

The overall grid size is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the CELLHEIGHTMIN= and CELLWIDTHMIN= options.

The START= option affects how the columns are populated.

The PANELNUMBER= option enables you to create multiple smaller grids that completely partition the classifier values.

COLUMNWEIGHT=EQUAL | PROPORTIONAL

specifies how weights are assigned to the columns widths.

EQUAL

all columns have equal width.

PROPORTIONAL

each column has a width that is proportional to the number of discrete midpoint values that it contains.

Restriction

At least one column axis must be discrete in order for PROPORTIONAL to have any effect. Otherwise, EQUAL is used.

Interactions

When COLUMNDATARANGE=UNIONALL, PROPORTIONAL is ignored and EQUAL is used.

When PROPORTIONAL is in effect, COLUMNDATARANGE=AUTO is interpreted as UNION.

If all of the following conditions are true, then the discrete axis is used to proportion the columns: PROPORTIONAL is in effect, both the X and X2 axes are used, and only one of the two axes is discrete. If both axes are discrete, then the X axis is used to proportion the columns.

When PROPORTIONAL is in effect, the OFFSETMIN= and OFFSETMAX= axis options are ignored in COLUMNAXISOPTS= and COLUMN2AXISOPTS=. The axis offsets are always set to one-half of the midpoint spacing regardless of plot type.

Default

EQUAL

HEADERBACKGROUND COLOR=style-reference | color

specifies the background color of the cell headers.

style-reference

specifies a style reference in the form style-element:style-attribute.

Only the style attribute named COLOR or CONTRASTCOLOR is used.

Default

The GraphHeaderBackground:Color style reference.

Interaction

HEADEROPAQUE= TRUE must be in effect in order for the color to be seen.
HEADERBORDER=TRUE | FALSE
specifies whether a border is drawn around the header cells.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default  TRUE

Tip  The border attributes are controlled by the GraphBorderLines style element.

See  "boolean" for other Boolean values that you can use.

HEADERLABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the data labels.

Default  The GraphValueText style element.

See  "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Text Options" on page 1525 for available text-options.

HEADERLABELDISPLAY=NAMEVALUE | VALUE | NONE
specifies the content of the cell headers.

NAMEVALUE
 displays the classification variable name and value as a name=value pair in each cell header.

Example  If the classification variables are Country and Product, then HEADERLABEL=NAMEVALUE produces cell headers such as the following:

  Country=CANADA
  Product=TABLE

VALUE
 displays the classification variable value only in each cell header.

Example  If the classification variables are Country and Product, then HEADERLABEL=VALUE produces cell headers such as the following:

  CANADA
  TABLE

NONE
 suppresses the cell headers.

Default  NAMEVALUE

HEADERLABELLOCATION=OUTSIDE | INSIDE
indicates whether the cell header is placed within each cell (INSIDE) or as row and column headers external to the lattice (OUTSIDE).

Default  OUTSIDE
HEADEROPAQUE=TRUE | FALSE
specifies whether the background for cell headers is opaque (TRUE) or transparent (FALSE).

Default: TRUE

Interaction: When this option is set to FALSE, the background color for cell headers is not used.

See “boolean” on page 1479 for other Boolean values that you can use.

HEADERPACK=TRUE | FALSE
specifies whether the header cells are consolidated into a comma-separated list in order to save space.

Note: This option is valid starting in SAS 9.4M1.

The following figure shows the effect of HEADERPACK= on the cell headers in one row of a data lattice. The data lattice contains classification variables Country and Year.

![HEADERPACK=FALSE](image)

![HEADERPACK=TRUE](image)

The following figure shows the effect of HEADERPACK= on the cell headers in one row of a data lattice. The data lattice contains classification variables Country, Year, and Product.

Default: FALSE

Note: If the length of the cell header text exceeds the available width, then the text is truncated.

Tip: If truncation occurs, then use the HEADERSPLITCOUNT= option to split the cell header text into multiple lines.

See “boolean” on page 1479 for other Boolean values that you can use.

HEADERSEPARATOR=“string”
specifies one or more characters to place between each value in the cell header when HEADERPACK=TRUE.

Note: This option is valid starting in SAS 9.4M1.
A comma followed by a space

This option is ignored when HEADERPACK=FALSE.

**HEADERSPLITCOUNT=positive-integer**

specifies the number of headers to consolidate on a header line before splitting the text to the next line.

Note: This option is valid starting in SAS 9.4M1.

The following figure shows how HEADERSPLITCOUNT=2 splits the cell header value in a data panel of classification variables Country, Year, and Product.

The cell header text is not split

This option is ignored when HEADERPACK=FALSE.

Note: If the length of the cell header text exceeds the available width, then the text is truncated.

Tip: Use the HEADERSEPARATOR= option to specify a different separator.

**INCLUDEMISSINGCLASS=TRUE | FALSE**

specifies whether to include grid cells for crossings of the ROWVAR and COLUMNVAR variables that contain a missing value.

**TRUE**

any crossing of the class variables that includes a missing value produces a row or column of cells in the grid.

**FALSE**

any crossing of the class variables that includes a missing value does not produce a row or columns of cells in the grid.

By default, missing class values are included in the classification. When the data contains missing classification values, cells are created for the missing classes. The classification headers for the missing values are blank for missing string values or a dot for missing numeric values. You can use the INCLUDEMISSINGCLASS=FALSE option to exclude the missing values. If you want to keep the missing values, then you can create a format that specifies more meaningful headings for the missing classes. For an example, see “Missing Class Values” in *SAS Graph Template Language: User’s Guide*.

Note: ODS Graphics does not support Unicode values in user-defined formats in SAS 9.4M2 and in earlier releases. Starting with SAS 9.4M3, ODS Graphics supports Unicode values in user-defined formats only if they are preceded by the (*ESC*) escape sequence. Example: "(*ESC*){unicode beta}". ODS Graphics does not support an escape character that is defined in an ODS ESCAPECHAR statement in user-defined formats.

**Default** TRUE

See “boolean ” on page 1479 for other Boolean values that you can use.
INSET=(variable-list)
specifies what information is displayed in an inset. The variable-list defines one or more variables whose names and values appear as a small table in the data cells. The variables can be either numeric or character. Variable names are separated by spaces.

Requirement No predefined information is available for the inset. You must create the desired inset information as part of your input data. See “Creating Your Inset Data” on page 71.

Note The variable values are associated with the data cells by data order. That is, the first observation from all the variables in variable-list are used in the first data cell, the second observation from all variables in variable-list are used in the second data cell, and so on. If a value is missing for an observation, then the corresponding name-value pair is skipped in the affected data cell.

Tip The location and appearance of the inset is controlled by the INSETOPTS= option.

See “Adding Insets to Your Graph” in SAS Graph Template Language: User’s Guide

INSETOPTS=(appearance-options)
specifies location and appearance options for the inset information. The appearance options can be any one or more of the following values:

AUTOALIGN=NONE | AUTO | (location-list)
specifies whether the inset is automatically aligned within the layout.

NONE
does not automatically align the inset.

This inset’s position is set by the HALIGN= and VALIGN=appearance-options.

AUTO
attempts to center the inset in the area that is farthest from any surrounding markers.

Data cells might have different inset placements.

(location-list)
restricts the inset’s possible locations to those locations in the specified location-list, and uses the location-list position that least collides with the data cell’s other graphics features.

The location-list is a space-separated list that can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Example AUTOALIGN=(TOPRIGHT TOLEFT)

Default NONE

Interaction When AUTOALIGN=AUTO or (location-list), the HALIGN= and VALIGN= options are ignored.

BACKGROUND COLOR=style-reference | color
specifies the color of the inset background.
**style-reference**
specifies a style reference in the form `style-element : style-attribute`.

Only the COLOR and CONTRASTCOLOR style attributes are valid.

**Default**
The background is transparent. No color is assigned.

**BORDER=TRUE | FALSE**
specifies whether a border is displayed around the inset.

**Default**
FALSE

**See**  
“boolean ” on page 1479 for other Boolean values that you can use.

**CONTENTDISPLAY=LABELVALUE | VALUE**
specifies whether the variable information that is displayed in the inset includes the column label and value, or only the column value.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

If a label is not assigned to a column, then the column name is used as the label for that column. Consider the following inset data:

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>TYPE</em></th>
<th>Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SS1</td>
<td>94.359</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

The following figure shows the effect that the CONTENTDISPLAY= option has on the content of an inset that displays this data.

<table>
<thead>
<tr>
<th>LABELVALUE</th>
<th>LABELVALUE with SEPARATOR=“=”</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>F Value 94.359</td>
<td>F Value = 94.359</td>
<td>94.359</td>
</tr>
<tr>
<td>Pr &gt; F &lt;.0001</td>
<td>Pr &gt; F = &lt;.0001</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

**Default**
LABELVALUE

**Tip**
Use the SEPARATOR= option to specify a separator other than the default blank space.

**DATASCHEME=LIST | MATCHED**
specifies the scheme that was used to merge the inset information into the analysis data.

**LIST**
one-to-one merging (no BY statement) was used to merge the inset and analysis data.

The variable values are associated with the cells of the data lattice by using data order. That is, the inset variable values in the first observation are used in the inset for the first cell, the inset variable values in the second observation are used in the inset for the second cell, and so on.

**MATCHED**
match-merging (using a BY statement) was used to merge the inset and analysis data.
Defaults

<table>
<thead>
<tr>
<th>List if the data is a SAS data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCHED if the data is a CAS in-memory table</td>
</tr>
</tbody>
</table>

**Tip**

MATCHED is the preferred data scheme for merging the inset and analysis data.

**See**

"Adding Insets to Classification Panels" in SAS Graph Template Language: User's Guide

HALIGN=LEFT | CENTER | RIGHT

specifies the horizontal alignment of the inset.

**Default**

LEFT

**Interaction**

This option has an effect only when this layout is nested within a REGION layout or when this layout is nested in an overlay-type layout and AUTOALIGN=NONE.

**OPAQUE=TRUE | FALSE**

specifies whether the inset background is opaque (TRUE) or transparent (FALSE).

**Default**

FALSE

**Interaction**

When OPAQUE=FALSE, the background color is not used.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**SEPARATOR="string"**

specifies a new separator for the column label and value.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**Default**

A blank space

**Interaction**

This option is ignored when CONTENTDISPLAY=VALUE.

**TEXTATTRS=style-element | style-element (text-options) | (text-options)**

specifies the text properties of the entire inset, excluding the title.

**Default**

The GraphDataText style element.

**See**

"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Text Options" on page 1525 for available text-options.

**TITLE="string"**

specifies a title for the inset.

The title is added at the top of the inset and spans the full inset width.

**Note**

Space is not reserved for the title when this value is not specified.

**Tip**

Text properties for the title string can be specified with TITLEATTRS=. 
TITLEATTRS=\textit{style-element} | \textit{style-element} (\textit{text-options}) | (\textit{text-options})
specifies the text properties of the inset’s title string.

\textbf{Default}  The GraphValueText style element.

\textbf{See}  "General Syntax for Attribute Options" on page 1519 for the syntax on using a \textit{style-element}.

\textit{Text Options} on page 1525 for available \textit{text-options}.

VALIGN=\textit{TOP} | \textit{CENTER} | \textit{BOTTOM}  
\textbf{specifies the vertical alignment of the inset.}

\textbf{Default}  \textit{TOP}

\textbf{Interaction}  This option has effect only when this layout is nested within a REGION layout or when this layout is nested in an overlay-type layout and AUTOALIGN=NONE.

\textbf{Requirements}  The options must be enclosed in parentheses.

Each option must be specified as a \textit{name} = \textit{value} pair, and each pair must be separated by a space.

\textbf{OPAQUE=\textit{TRUE} | \textit{FALSE}}  
\textbf{specifies whether the layout background is opaque (\textit{TRUE}) or transparent (\textit{FALSE}).}

\textbf{Default}  \textit{FALSE}

\textbf{Interaction}  When this option is set to \textit{FALSE}, the background color is not used.

\textbf{See}  "\textit{boolean }" on page 1479 for other Boolean values that you can use.

\textbf{OUTERPAD=\textit{AUTO} | dimension | (pad-options)}  
\textbf{specifies the amount of extra space to add outside the layout border.}

\textbf{AUTO}  
specifies that the default outside padding for this component is used.

\textit{dimension}  
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

\textbf{(pad-options)}  
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can contain one or more of the following pad options:

\textbf{LEFT=}\textit{dimension}  
specifies the amount of extra space to add to the left side.

\textbf{RIGHT=}\textit{dimension}  
specifies the amount of extra space to add to the right side.

\textbf{TOP=}\textit{dimension}  
specifies the amount of extra space to add to the top.

\textbf{BOTTOM=}\textit{dimension}  
specifies the amount of extra space to add to the bottom.
Note Sides that are not assigned padding are padded with the default amount.

Tip Use pad-options to create non-uniform padding.

Default AUTO

Note The default units for dimension are pixels.

See “dimension” on page 1480

**PAD=dimension | (pad-options)**

specifies the amount of extra space that is added inside the layout border.

*dimension*

specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

*(pad-options)*

a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

**LEFT=dimension**

specifies the amount of extra space added to the left side.

Default 0

**RIGHT=dimension**

specifies the amount of extra space added to the right side.

Default 0

**TOP=dimension**

specifies the amount of extra space added to the top.

Default 0

**BOTTOM=dimension**

specifies the amount of extra space added to the bottom.

Default 0

Note Sides that are not assigned padding are padded with the default amount.

Tip Use pad-options to create non-uniform padding.

Note The default units for dimension are pixels.

See “dimension” on page 1480

**PANELNUMBER=positive-integer**

specifies the number of the panel to produce. This option enables you to partition a large grid into a number of smaller grids under these conditions:

- You set a grid size explicitly (ROWS= and COLUMNS= when ROWVAR and COLUMNVAR variables are specified; ROWS= when only a ROWVAR
variable is specified; COLUMNS= when only a COLUMNVAR variable is specified)

- The grid size is smaller in one or both of the dimensions of the default dynamically generated grid.
- You execute the template N times and increment the panel number each time. N is determined by CEIL(all rows * all columns / grid rows * grid columns).

Default 1

Example Suppose ROWVAR=R (R has 10 unique values) and COLUMNVAR=C (C has 11 unique values). The dynamic grid has 10 rows and 11 columns and you would have to make the HEIGHT=and WIDTH= quite large to enable 110 plots to be displayed. By setting some smaller grid size, say ROWS=3 and COLUMNS=4, and by making the value of PANELNUMBER= a dynamic or macro variable, you can create 10 panels (9 with 12 data cells and 1 with 2 data cells) that collectively display all 110 possible crossings. You simply invoke PROC SGRENDER or a DATA step 10 times, incrementing the dynamic value for PANELNUMBER each time.

ROWAXISOPTS=(axis-options)
specifies Y-axis options for all rows.

Requirement Axis options must be enclosed in parentheses and separated by spaces.

See “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” on page 1149 for a list of options.

ROW2AXISOPTS=(axis-options)
specifies Y2-axis options for all rows.

Requirement Axis options must be enclosed in parentheses and separated by spaces.

Interaction This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement's YAXIS= option to map data to the Y2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974

See “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” on page 1149 for a list of options.

ROWDATARANGE=AUTO | UNIONALL | UNION
specifies how the Y-axes of instances of the graph-prototype are scaled.

AUTO automatically selects the Y-axis scale based on the ROWWEIGHT= option and the column axis type.

The Y-axis scale is selected as follows:

- When ROWWEIGHT=EQUAL (default), UNIONALL is selected.
- When ROWWEIGHT=PROPORTIONAL and the row axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.
UNIONALL
scales the Y-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

UNION
scales the Y-axis data ranges separately for each row in the layout on a per-panel basis.
The scaling does not span multiple panels.

Default AUTO

Tip Use the ROWAXISOPTS= option to control shared axis features.

See The ROWWEIGHT= option.
The PANELNUMBER= option for information about how to create multiple panels.

ROW2DATARANGE=AUTO | UNIONALL | UNION
specifies how the Y2-axes of instances of the graph-prototype are scaled.

AUTO automatically selects the Y2-axis scale based on the ROWWEIGHT= option and the column axis type.
The Y2-axis scale is selected as follows:
- When ROWWEIGHT=EQUAL (default), UNIONALL is selected.
- When ROWWEIGHT=PROPORTIONAL and the row axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

UNIONALL
scales the Y2-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

UNION
scales the Y2-axis data ranges separately for each row in the layout on a per-panel basis.
The scaling does not span multiple panels.

Default AUTO

Interaction This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement’s YAXIS= option to map data to the Y2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974

Tip Use the ROW2AXISOPTS= option to control shared axis features.

See The ROWWEIGHT= option.
The PANELNUMBER= option for information about how to create multiple panels.

ROWGUTTER=dimension
specifies the amount of empty space between the rows.

Default 0
Note: If there are \( n \) rows, then there are \( n-1 \) gutters.

See: "dimension" on page 1480

**ROWHEADERS=RIGHT | LEFT | BOTH**

specifies where to position the outside row heading.

**RIGHT**

specifies that row heading appears at the right of the layout.

**LEFT**

specifies that row heading appears at the left of the layout.

**BOTH**

specifies that row heading alternates between the right and left of the layout row by row.

Default: **RIGHT**

Requirement: **HEADERLABELLOCATION= OUTSIDE** must be set for this option to have any effect.

**ROWS=integer**

specifies the number of rows in the layout.

Defaults: If this option is not specified, then the number of rows is dynamically adjusted to equal the number of classifier values for the ROWVAR= variable.

If this option is specified, then the specified number of rows is created. If the number of ROWVAR classifier values is greater than the specified number of rows, then no graph is created for some classifier values. If the number of classifier values is smaller than the specified number of rows, then extra empty rows are created.

Interactions: The overall grid size is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the \( \text{CELLHEIGHTMIN=} \) and \( \text{CELLWIDTHMIN=} \) options.

The **START=** option affects how the rows are populated.

Tip: The **PANELNUMBER=** option enables you to create multiple smaller grids that completely partition the classifier values.

**ROWWEIGHT=EQUAL | PROPORTIONAL**

specifies how weights are assigned to the row heights.

**EQUAL**

all rows have equal height.

**PROPORTIONAL**

each row has a height that is proportional to the number of discrete midpoint values that it contains.

Restriction: At least one row axis must be discrete in order for PROPORTIONAL to have any effect. Otherwise, EQUAL is used.
Interactions

When ROWDATARANGE=UNIONALL, PROPORTIONAL is ignored and EQUAL is used.

When PROPORTIONAL is in effect, ROWDATARANGE=AUTO is interpreted as UNION.

If all of the following conditions are true, then the discrete axis is used to proportion the rows: PROPORTIONAL is in effect, both the Y and Y2 axes are used, and only one of the two axes is discrete. When both axes are discrete, the Y axis is used to proportion the rows.

When PROPORTIONAL is in effect, the OFFSETMIN= and OFFSETMAX= axis options are ignored in ROWAXISOPTS= and ROW2AXISOPTS=. The axis offsets are always set to one-half of the midpoint spacing regardless of plot type.

Default  EQUAL

**SHRINKFONTS=TRUE | FALSE**

specifies whether fonts in the layout are scaled down.

Default  FALSE

Note

If SHRINKFONTS=TRUE, the font scaling applies to all child layouts within the parent layout. If a child layout also specifies SHRINKFONTS=TRUE, the fonts for the children of that child layout are further scaled down.

See  "boolean " on page 1479 for other Boolean values that you can use.

**SKIPEMPTYCELLS=TRUE | FALSE**

specifies whether the external axes skip the empty cells in a partially filled grid.

**TRUE**

skips empty cells and "snaps" the external axes to the nearest data cell, both vertically and horizontally.

Though the empty cells are not displayed, the data cells in the grid are not enlarged to fill the area.

**FALSE**

displays external axes at their normal locations, even if there are empty cells at one or more of the locations.

Whenever the number of unique COLUMNVAR= classifier values (data cells) is not evenly divisible by the COLUMNS= value, or the number of unique ROWVAR= classifier values (data cells) is not evenly divisible by the ROWS= value, then one or more panels is partially filled with data cells and padded with empty cells to complete the grid.

Here is an example of a data lattice that consists of 4 column-data cells and 3 row-data cells arranged in a 4-column, 2-row grid. The following figure shows the default appearance of the last panel:
When SKIPEMPTYCELLS=TRUE, the empty padding cells of all panels are removed and external axis ticks and tick values snap to the data cells:

![Graph](image)

Default: FALSE

See "boolean" on page 1479 for other Boolean values that you can use.

SORTORDER=(COLUMNVAR=sort-order-keyword | ROWVAR=sort-order-keyword)
specifies the order of the cells along the columns and rows.

COLUMNVAR
the column role.

ROWVAR
the row variable role.

sort-order-keyword
a sort-order keyword.

The keyword can be one of the following:

AUTO
sorts using DATA for character data and ascending unformatted for numeric data.

DATA
retains the data order.

Note: In SAS 9.4M6 and SAS 9.4M5, when data stored in a CAS in-memory table is used and DATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDINGFORMATTED or DESCENDINGFORMATTED. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in
a CAS in-memory table is used and DATA is specified, DATA is ignored, and AUTO is used instead.

**ASCENDING FORMATTED**
sorts in ascending order, using the formatted values.

**DESCENDING FORMATTED**
sorts in descending order, using the formatted values.

**Defaults**
AUTO for all roles if the data is a SAS data set
ASCENDING FORMATTED for all roles if the data is a CAS in-memory table

**Tip**
The placement of the cells within the layout also depends on the starting location, which is controlled by the START= option.

**START=TOP LEFT | BOTTOM LEFT**
indicates whether to start populating the grid from the top left or bottom left corner. If ROWVAR=R has values in the sort order 1 and 2 and COLUMNVAR=C has values in the sort order a and b, then START=BOTTOM LEFT is populated as follows:

```
plot plot | R=2
plot plot | R=1
C=a  C=b
```

START=TOP LEFT is populated as follows:

```
plot plot | R=1
plot plot | R=2
C=a  C=b
```

**Default**
TOP LEFT

**SIDEBAR Statement Options**

**ALIGN= BOTTOM | TOP | LEFT | RIGHT**
specifies the sidebar’s location within the layout. You can specify up to four SIDEBAR blocks in a LAYOUT DATALATTICE, one for each of the bottom, top, left, and right sidebar positions.

- The LAYOUT DATALATTICE automatically aligns a sidebar with the layout columns or rows.
- Only one statement (such as ENTRY or DISCRETELEGEND) or one layout block (such as LAYOUT GRIDDED) is allowed in a SIDEBAR block. To create multi-line text in a sidebar, nest ENTRY statements within a LAYOUT GRIDDED block.

**Default**
BOTTOM
SPACEFILL=TRUE | FALSE
specifies whether to fill all the area of the sidebar with its contents.

- Default: TRUE

Tip: To prevent a layout block within the sidebar from expanding to the sidebar boundaries, set this option to FALSE.

See "boolean" on page 1479 for other Boolean values that you can use.

Details

Statement Description

The LAYOUT DATALATTICE statement makes it easy to create a grid of graphs, based on the values of one or two classifications variables. To create a grid that is based on more than two classification variables, or to have more control over the grid layout, use LAYOUT DATAPANEL instead.

By default, the number of cells in the layout is determined by the number of value pairings that are possible for the classification values plus any empty cells needed to complete the last row or column of the grid. The contents of each data cell are based on a graph prototype that you specify in the graph-prototype-block. You can enhance the display using one or more sidebar-statement-blocks. For classification variables that have many values, you can use the COLUMNS= and or ROWS= options and the PANELNUMBER= option to generate multiple panel displays.

Classification variables for the layout are specified on the ROWVAR= argument (to specify a row variable), or the COLUMNVAR= argument (to specify a column), or both arguments to specify both a column and a row variable. The graph prototype for each data cell’s contents is specified within a "LAYOUT PROTOTYPE Block" on page 71 block, and sidebars are specified within SIDEBAR blocks. The LAYOUT PROTOTYPE and SIDEBAR blocks are nested within the LAYOUT DATALATTICE block.

By default, the first data cell to be filled is in the layout’s top left corner. Use the START= option to change the starting data cell to the bottom left corner.

Rather than display the header labels outside the grid, you can set HEADERLABELLOCATION= INSIDE to display them inside the grid, as shown in the following figure:
The `DATALATTICE` layout is designed to be the outermost layout in the template.

Managing Rows and Columns

If you do not explicitly manage columns and rows using the `COLUMNS=` and `ROWS=` options, then the default layout behavior is as follows:

- If both `ROWVAR=` and `COLUMNVAR=` are specified, then a data cell is created for each of the value pairings that are possible for the classification values of the specified variables. If the `ROWVAR` variable has R distinct values and the `COLUMNVAR` variable has C distinct values, then the dimension of grid produced is R x C.

- If only the `ROWVAR` variable is used, then an R x 1 grid is produced.

- If only the `COLUMNVAR` variable is used, then a 1 x C grid is produced.

If the class variable is of type character, then its values are returned in data order. To control the ordering of the values, you can sort the input data by the classification variables. If the class variable is of type numeric, then the values are displayed in ordinal order.

Formats can be assigned to class variables to create classification levels (for example, an `AGEGROUPFMT` format for numeric `AGE`). In this case, the classification is performed after the format is applied. For numeric data, the order is ordinal, based on the first value in each class.

Use the `INCLUDEMISSINGCLASS` option to control whether cells are displayed when any value crossing contains a missing value.

The output size does not grow automatically as the number of cells increases. To set a panel size for the current template, use the `DESIGNHEIGHT=` and `DESIGNWIDTH=` options in the `BEGINGRAPH` statement. To set a panel size for all templates in the current SAS session, use the `HEIGHT=` and `WIDTH=` options in the...
ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement. The default output width is 640px, and the default output height is 480px.

As the number of cells in the grid increases, the size of each cell decreases. At some point the cells might become so small that a meaningful graph cannot be rendered. The CELLHEIGHTMIN= and CELLWIDTHMIN= options set a threshold for the smallest cell. If the actual cell height or width becomes smaller, then no panel is drawn. The default minimum cell size is CELLHEIGHTMIN=100px and CELLWIDTHMIN=100px.

Using the default panel size and cell size, the DATALATTICE layout accommodates a grid of about 24 cells (6 columns by 4 rows). If you know that the number of cells is larger, then you should increase the overall panel size, or decrease the minimum cell size, or both. You can also use ROWS=, COLUMNS=, and PANELNUMBER= options to partition your data so that a number of smaller grids are produced that cumulatively show all of the value crossings.

Creating Your Inset Data

When you use the INSET= option to insert an inset, no predefined information is available for the inset. You must create the desired inset information as part of your input data. This is most typically done as follows:

- Create a separate data set for the inset columns making sure that the column names are different from the other columns used in graph. The number observations of inset data should match the number of cells in the classification panel. The ordering of the inset observations should be the same as the population order of the classification panel’s cells, taking into account the ROWVAR= and COLUMNVAR= arguments and the START= option. Typically, the number of observations for the inset data is smaller than the other input data for the graph.

- Merge the inset data set with the data set for the graph using a DATA or PROC SQL step. Do not match-merge the observations of the two data sets (no BY processing). The resulting data set typically has the inset columns padded with missing values.

- Use the merged data set to produce the graph, specifying the inset column names in this option’s variable-list.

LAYOUT PROTOTYPE Block

You must specify a single LAYOUT PROTOTYPE block within the LAYOUT DATALATTICE block, using the following syntax:

```
LAYOUT PROTOTYPE </options>

GTL-statement(s);

ENDLAYOUT;
```

The LAYOUT PROTOTYPE block determines the graphical content of each data cell and is repeated within each data cell, based on the subsets of the classification variables.

For more information about the LAYOUT PROTOTYPE block and the list of available options, see “LAYOUT PROTOTYPE Statement” on page 173.
SIDEBAR Blocks

A LAYOUT DATALATTICE enables you to display one or more sidebars outside of the axes. A sidebar spans across columns or rows and is useful for displaying information that applies to all of the columns or all of the rows. For example, sidebars are useful for displaying a legend.

A SIDEBAR statement has the following syntax:

```
SIDEBAR <options>
    GTL-statement(s);
ENDSIDEBAR;
```

The following example shows a SIDEBAR block that displays a legend at the top of the layout grid.

```
sidebar / align=top;
    discretelegend 'p' 'a' / across=2;
endsidebar;
```

Example: LAYOUT DATALATTICE

This example shows the result of using row and column classification variables. In this case, a four-column, three-row data lattice is created:

- The classification values are placed as row or column labels by default.
- The ROWDATARANGE=UNION option assures that an axis range is computed separately for each row using the data ranges of all \texttt{Y=} columns in that row. This facilitates the visual comparison of the data cells.
- A SIDEBAR block is used to place the legend at the bottom of the lattice.

Example Graph

The following graph was generated by the "Example Program" on page 73:
Example Program

```sas
proc template;
  define statgraph layoutdatalattice;
  begingraph;
    entrytitle "Annual Furniture Sales Comparisons";
    layout datalattice rowvar=country columnvar=year /
      rowdatarange=union
      headerlabeldisplay=value
      headerbackgroundcolor=GraphAltBlock:color
      rowaxisopts=(display=(tickvalues) griddisplay=on
        linearopts=(tickvalueformat=dollar12.))
      columnaxisopts=(display=(tickvalues)
        timeopts=(tickvalueformat=monname3.));
    layout prototype / cycleattrs=true;
      seriesplot x=month y=TotalActual / name="Actual";
      seriesplot x=month y=TotalPredict / name="Predict";
    endlayout;
    sidebar / align=bottom;
      discretelegend "Actual" "Predict" / border=false;
    endsidebar;
  endlayout;
  endgraph;
end;
run;

proc summary data=sashelp.prdsal2 nway;
  class country year month;
  var actual predict;
```
output out=prdsal2 sum=TotalActual TotalPredict;
run;
proc sgrender data=prdsal2 template=layoutdatalattice;
run;

LAYOUT DATAPANEL Statement

Creates a grid of graphs based on one or more classification variables and a graphical prototype. By
default, a separate instance of the prototype (a data cell) is created for each actual combination of the
classification variables.

Restriction: You can specify only one LAYOUT PROTOTYPE block in the LAYOUT DATAPANEL
block. If you specify more than one, then only the last prototype block specified is
honored. The remaining prototype blocks are ignored.

Tip: The DATAPANEL layout should be the outermost layout in the template.

Syntax

\[
\text{LAYOUT DATAPANEL CLASSVARS=}(\text{class-var1\ldots class-varN}) \langle/\text{options}\rangle; \\
\text{LAYOUT PROTOTYPE} \langle/\text{options}\rangle; \\
\text{GTL-statements}; \\
\text{ENDLAYOUT}; \\
\langle/\text{SIDEBAR} \langle/\text{options}\rangle; \\
\text{GTL-statement(s)}; \\
\text{ENDSIDEBAR}; \\
\langle/\text{... more-sidebar-statement-blocks ...} \rangle \\
\text{ENDLAYOUT};
\]

Summary of Optional Arguments

Appearance options

- **BACKGROUNDCOLOR=style-reference | color**
  specifies the color of the layout background.

- **BORDER=TRUE | FALSE**
  specifies whether a border is drawn around the layout.

- **BORDERATTRS=style-element | style-element (line-options) | (line-options)**
  specifies the attributes of the border line around the layout.

- **CELLHEIGHTMIN=dimension**
  specifies the minimum height of a cell in the grid.

- **CELLWIDTHMIN=dimension**
  specifies the minimum width of a cell in the grid.

- **COLUMNGUTTER=dimension**
  specifies the amount of empty space that is between the columns.

- **HEADERBACKGROUNDCOLOR=style-reference | color**
  specifies the background color of the cell headers.

- **HEADERBORDER=TRUE | FALSE**
  specifies whether a border is drawn around the header cells.
HEADERLABELATTRS=\text-element \text-element \text-options \text-options \text-options
specifies the color and font attributes of the data labels.

HEADEROPAQUE=\text{TRUE} \text{FALSE}
specifies whether the background for cell headers is opaque (TRUE) or transparent (FALSE).

HEADERPACK=\text{TRUE} \text{FALSE}
specifies whether the header cells are consolidated into a comma-separated list in order to save space.

HEADERSEPARATOR="\text{string}"
specifies one or more characters to place between each value in the cell header when HEADERPACK=TRUE.

HEADERSPLITCOUNT=\text{positive-integer}
specifies the number of headers to consolidate on a header line before splitting the text to the next line.

OPAQUE=\text{TRUE} \text{FALSE}
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

OUTERPAD=\text{AUTO} \text{dimension} \text{pad-options}
specifies the amount of extra space to add outside the layout border.

PAD=\text{dimension} \text{pad-options}
specifies the amount of extra space that is added inside the layout border.

SHRINKFONTS=\text{TRUE} \text{FALSE}
specifies whether fonts in the layout are scaled down.

SORTORDER=(\text{COLUMNVAR}=\text{sort-order-keyword} \text{ROWVAR}=\text{sort-order-keyword})
specifies the order of the cells along the columns and rows.

\textbf{Axis options}

\text{COLUMN2AXISOPTS}=\text{axis-options}
specifies X2-axis options for all columns.

\text{COLUMN2DATARANGE}=\text{AUTO} \text{UNIONALL} \text{UNION}
specifies how the X2-axes of instances of the graph-prototype are scaled.

\text{COLUMNAXISOPTS}=\text{axis-options}
specifies X-axis options for all columns.

\text{COLUMNNDATARANGE}=\text{AUTO} \text{UNIONALL} \text{UNION}
specifies how the X-axes of instances of the graph-prototype are scaled.

\text{ROW2AXISOPTS}=\text{axis-options}
specifies Y2-axis options for all rows.

\text{ROW2DATARANGE}=\text{AUTO} \text{UNIONALL} \text{UNION}
specifies how the Y2-axes of instances of the graph-prototype are scaled.

\text{ROWAXISOPTS}=\text{axis-options}
specifies Y-axis options for all rows.

\text{ROWDATARANGE}=\text{AUTO} \text{UNIONALL} \text{UNION}
specifies how the Y-axes of instances of the graph-prototype are scaled.

\textbf{Inset options}

\text{INSET}=\text{variable-list}
specifies what information is displayed in an inset.

\text{INSETOPTS}=\text{appearance-options}
specifies location and appearance options for the inset information.
Layout options

**COLUMN=** integer
specifies the number of columns in the layout.

**COLUMNWEIGHT=** EQUAL | PROPORTIONAL
specifies how weights are assigned to the columns widths.

**HEADERLABELDISPLAY=** NAMEVALUE | VALUE | NONE
specifies the content of the cell headers.

**INCLUDEMISSINGCLASS=** TRUE | FALSE
specifies whether to include grid cells for crossings of the CLASSVARS variables that contain a missing value.

**ROWGUTTER=** dimension
specifies the amount of empty space between the rows.

**ROWS=** integer
specifies the number of rows in the layout.

**ROWWEIGHT=** EQUAL | PROPORTIONAL
specifies how weights are assigned to the row heights.

Panel options

**ORDER=** ROWMAJOR | COLUMNMAJOR
specifies whether data cells are populated by column priority or by row priority.

**PANELNUMBER=** positive-integer
specifies the number of the panel to produce.

**SKIPEMPTYCELLS=** TRUE | FALSE
specifies whether the external axes skip the empty cells in a partially filled grid.

**SPARSE=** TRUE | FALSE
specifies whether crossings of the class variables include only the crossings in the data or all possible crossings.

**START=** TOPLEFT | BOTTOMLEFT
indicates whether to start populating the grid cells from the top left or bottom left corner.

Role options

**ROLENAMES=** (role-name-list)
specifies user-defined roles for information contained in data columns.

Required Argument

**CLASSVARS=** (column-list)
specifies a list of classification variables. By default, a data cell is created for each crossing of these variables in the input data. The total number of grid cells created is the result of a crosstabulation table of all the classification variables plus any empty cells needed to complete the last row or column of the grid. You can request that data cells be generated for all possible crossings, even when the class variables have no values at those crossings. For more information, see the **SPARSE=** option.

If the class variable is of type character, then its values are returned in data order. To control the ordering of the values, you can sort the input data by the classification variables. If the class variable is of type numeric, then the values are displayed in ordinal order.
Formats can be assigned to class variables to create classification levels (for example, an AGEGROUPFMT format for numeric AGE). In this case, the classification is performed after the format is applied. For numeric data, the order is ordinal, based on the first value in each class.

Use the `INCLUDEMISSINGCLASS` option to control whether cells are displayed when any value crossing contains a missing value.

The output size does not grow automatically as the number of cells increases. To set a panel size for the current template, use the `DESIGNHEIGHT=` and `DESIGNWIDTH=` options in the `BEGINGRAPH` statement. To set a panel size for all templates in the current SAS session, use the `HEIGHT=` and `WIDTH=` options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the `BEGINGRAPH` statement. The default output width is 640px, and the default output height is 480px.

As the number of cells in the grid increases, the size of each cell decreases. At some point the cells might become so small that a meaningful graph cannot be rendered. The `CELLHEIGHTMIN=` and `CELLWIDTHMIN=` options set a threshold for the smallest cell. If the actual cell height or width becomes smaller, then no panel is drawn. The default minimum cell size is `CELLHEIGHTMIN=100px` and `CELLWIDTHMIN=100px`.

Using the default panel size and cell size, the DATAPANEL layout accommodates a grid of about 24 cells (6 columns by 4 rows). If you know that the number of cells is larger, then you should increase the overall panel size, or decrease the minimum cell size, or both. You can also use `ROWS=`, `COLUMNS=`, and `PANELNUMBER=` options to partition your data so that a number of smaller grids are produced that cumulatively show all of the value crossings.

Optional Arguments

**BACKGROUNDCOLOR=** `style-reference | color`

specifies the color of the layout background.

- `style-reference`
  specifies a style reference in the form `style-element:style-attribute`.
  Only the style attribute named `COLOR` or `CONTRASTCOLOR` is used.

- `color`
  specifies a color by name.

See `color` on page 1480

**Default**

The GraphBackground:Color style reference.

**Interaction**

`OPAQUE=TRUE` must be in effect in order for the color to be seen. By default, `OPAQUE=FALSE`.

**BORDER=** `TRUE | FALSE`

specifies whether a border is drawn around the layout.

**Default**

`FALSE`

**Interaction**

If this option is set to `FALSE`, then the `BORDERATTRS=` option is ignored.

See “`boolean`” on page 1479 for other Boolean values that you can use.
**BORDERATTRS=** *style-element | style-element* *(line-options)* | *(line-options)*  
specifies the attributes of the border line around the layout.

**Default**  
The GraphBorderLines style element.

**Interaction**  
**BORDER= TRUE** must be set for this option to have any effect.

**See**  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Line Options” on page 1522 for available *line-options*.

**HEADERBORDER=** *TRUE | FALSE*  
specifies whether a border is drawn around the header cells.

**Note:** This feature applies to **SAS 9.4M3** and to later releases.

**Default**  
**TRUE**

**Tips**  
The border attributes are controlled by the GraphBorderLines style element.

**HEADERBORDER=** *FALSE* does not remove the border around empty cells. If your panel has empty cells and you want to remove the cell borders, you can specify **SKIPEMPTYCELLS=** *TRUE* with **HEADERBORDER=** *FALSE* to remove the empty cells and the cell borders.

**See**  
“boolean” for other Boolean values that you can use.

**CELLHEIGHTMIN=** *dimension*  
specifies the minimum height of a cell in the grid. Use this option in conjunction with the **CELLWIDTHMIN=** option to set the minimum cell size.

The overall size of the panel is constrained by the **HEIGHT=** and **WIDTH=** options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum height threshold for all cells. If the actual cell height becomes smaller, then no panel is drawn.

**Default**  
100px

**See**  
“dimension” on page 1480

**CELLWIDTHMIN=** *dimension*  
specifies the minimum width of a cell in the grid. Use this option in conjunction with the **CELLHEIGHTMIN=** option to set the minimum cell size.

The overall size of the panel is constrained by the **HEIGHT=** and **WIDTH=** options in the ODS GRAPHICS statement. As the number of cells in the grid increases, the size of each cell decreases. At some point the cell becomes so small that a meaningful graph cannot be rendered. This option sets the minimum width threshold for all cells. If the actual cell width becomes smaller, then no panel is drawn.

**Default**  
100px
See “dimension” on page 1480

COLUMNAXISOPTS=(axis-options)
specifies X-axis options for all columns.

Restriction Axis options must be enclosed in parentheses and separated by spaces.

See “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” for a list of options.

COLUMN2AXISOPTS=(axis-options)
specifies X2-axis options for all columns.

Restriction This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement’s XAXIS= option to map data to the X2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974

Requirement Axis options must be enclosed in parentheses and separated by spaces.

See “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” for a list of options.

COLUMNDATARANGE=AUTO | UNIONALL | UNION
specifies how the X-axes of instances of the graph-prototype are scaled.

AUTO
selects the X-axis scale based on the COLUMNWEIGHT= option.

The X-axis scale is determined, as follows:

- When COLUMNWEIGHT=EQUAL (default), UNIONALL is selected.
- When COLUMNWEIGHT=PROPORTIONAL and the column axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

UNIONALL
scales the X-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION
scales the X-axis data ranges separately for each column on a per-panel basis.

The scaling does not span multiple panels.

Default AUTO

Tip Use the COLUMNAXISOPTS= option to control shared axis features.

See The COLUMNWEIGHT= option.

The PANELNUMBER= option for information about how to create multiple panels.

COLUMN2DATARANGE=AUTO | UNIONALL | UNION
specifies how the X2-axes of instances of the graph-prototype are scaled.
AUTO
automatically selects the X2-axis scale based on the COLUMNWEIGHT= option and the column axis type.

The X2-axis scale is selected as follows:
- When COLUMNWEIGHT=EQUAL (default), UNIONALL is selected.
- When COLUMNWEIGHT=PROPORTIONAL and the column axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

UNIONALL
scales the X2-axis data ranges across all layout columns and panels (when PANELNUMBER= is in effect).

UNION
scales the X2-axis data ranges separately for each column on a per-panel basis.

The scaling does not span multiple panels.

Default AUTO

Interaction
This option is needed only if you use a plot statement that supports a secondary X2 axis. If you do not use that statement’s XAXIS= option to map data to the X2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974

Tip
Use the COLUMN2AXISOPTS= option to control shared axis features.

See
The COLUMNWEIGHT= option.

The PANELNUMBER= option for information about how to create multiple panels.

COLUMNGUTTER=dimension
specifies the amount of empty space that is between the columns.

Default 0

Note
If there are n columns, then there are n-1 gutters.

See
“dimension” on page 1480

COLUMNS=integer
specifies the number of columns in the layout.

Defaults
If this option is not specified and ROWS= is specified, then the number of data cells (and columns) increases dynamically to allow all classifier values to be presented.

If both this option and ROWS= are specified, then a grid of that size is created, regardless of the number of classifier values. If the number of classifier values is greater than the grid size, then no graphs are created for some classifier values. If the number of classifier values is small and the grid size large, then there might be empty cells created.
The overall grid size is constrained by the \texttt{HEIGHT=} and \texttt{WIDTH=} options in the ODS GRAPhICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the \texttt{CELLHEIGHTMIN=} and \texttt{CELLWIDTHMIN=} options.

The \texttt{START=} and \texttt{ORDER=} options affect how the rows are populated.

The \texttt{PANELNUMBER=} option enables you to create multiple smaller grids that completely partition the classifier values.

\textbf{COLUMNWEIGHT=\texttt{EQUAL} | \texttt{PROPORTIONAL}}

specifies how weights are assigned to the columns widths.

\textbf{EQUAL}

all columns have equal width.

\textbf{PROPORTIONAL}

each column has a width that is proportional to the number of discrete midpoint values that it contains.

\textbf{Restriction}

At least one column axis must be discrete in order for \texttt{PROPORTIONAL} to have any effect. Otherwise, \texttt{EQUAL} is used.

\textbf{Interactions}

When \texttt{COLUMNDATARANGE=}\texttt{UNIONALL}, \texttt{PROPORTIONAL} is ignored and \texttt{EQUAL} is used.

When \texttt{PROPORTIONAL} is in effect, \texttt{COLUMNDATARANGE=}\texttt{AUTO} is interpreted as \texttt{UNION}.

If all of the following conditions are true, then the discrete axis is used to proportion the columns: \texttt{PROPORTIONAL} is in effect, both the X and X2 axes are used, and only one of the two axes is discrete. If both axes are discrete, then the X axis is used to proportion the columns.

When \texttt{PROPORTIONAL} is in effect, the \texttt{OFFSETMIN=} and \texttt{OFFSETMAX=} axis options are ignored in \texttt{COLUMNAXISOPTS=} and \texttt{COLUMN2AXISOPTS=}.

The axis offsets are always set to one-half of the midpoint spacing regardless of plot type.

\textbf{Default} \texttt{EQUAL}

\textbf{HEADERBACKGROUND\texttt{COLOR=}}\textit{style-reference} | \textit{color}

specifies the background color of the cell headers.

\textit{style-reference}

specifies a style reference in the form \textit{style-element}:\textit{style-attribute}.

Only the style attribute named \texttt{COLOR} or \texttt{CONTRASTCOLOR} is used.

\textbf{Default}

The \textit{GraphHeaderBackground}:\textit{Color} style reference.

\textbf{Interaction} \texttt{HEADEROPAQUE=} \texttt{TRUE} must be in effect in order for the color to be seen.
HEADERLABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the data labels.

Default: The GraphValueText style element.

See: “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

HEADERLABELDISPLAY=NAMEVALUE | VALUE | NONE
specifies the content of the cell headers.

NAMEVALUE
displays the classification variable name and value as a name=value pair in each cell header.

Example: If the classification variables are Country and Product, then HEADERLABEL=NAMEVALUE produces cell headers such as the following:

Country=CANADA
Product=TABLE

VALUE
displays the classification variable value only in each cell header.

Example: If the classification variables are Country and Product, then HEADERLABEL=VALUE produces cell headers such as the following:

CANADA
TABLE

NONE
suppresses the cell headers.

Default: NAMEVALUE

HEADEROPAQUE=TRUE | FALSE
specifies whether the background for cell headers is opaque (TRUE) or transparent (FALSE).

Default: TRUE

Interaction: When this option is set to FALSE, the background color for cell headers is not used.

See: “boolean ” on page 1479 for other Boolean values that you can use.

HEADERPACK=TRUE | FALSE
specifies whether the header cells are consolidated into a comma-separated list in order to save space.

Note: This option is valid starting in SAS 9.4M1.
The following figure shows the effect of HEADERPACK= on the cell headers in one row of a data lattice. The data lattice contains classification variables Country and Year.

The following figure shows the effect of HEADERPACK= on the cell headers in one row of a data lattice. The data lattice contains classification variables Country, Year, and Product.

```
<table>
<thead>
<tr>
<th>HEADERPACK=FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
</tr>
<tr>
<td>Canada</td>
</tr>
<tr>
<td>DESK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HEADERPACK=TRUE</th>
</tr>
</thead>
</table>
```

**Default**: FALSE

**Note**: If the length of the cell header text exceeds the available width, then the text is truncated.

**Tip**: If truncation occurs, then use the HEADERSPLITCOUNT= option to split the cell header text into multiple lines.

**See**: “boolean” on page 1479 for other Boolean values that you can use.

**HEADERSEPARATOR=** "string"

specifies one or more characters to place between each value in the cell header when HEADERPACK=TRUE.

**Note**: This option is valid starting in SAS 9.4M1.

**Default**: A comma followed by a space

**Interaction**: This option is ignored when HEADERPACK=FALSE.

**HEADERSPLITCOUNT=** positive-integer

specifies the number of headers to consolidate on a header line before splitting the text to the next line.

**Note**: This option is valid starting in SAS 9.4M1.

The following figure shows how HEADERSPLITCOUNT=2 splits the cell header value in a data panel of classification variables Country, Year, and Product.
The cell header text is not split

This option is ignored when `HEADERPACK=FALSE`.

If the length of the cell header text exceeds the available width, then the text is truncated.

Use the `HEADERSEPARATOR=` option to specify a different separator.

**INCLUDEMISSINGCLASS=TRUE | FALSE**

specifies whether to include grid cells for crossings of the `CLASSVARS` variables that contain a missing value.

**TRUE**

any crossing of the class variables that includes a missing value produces a row or column of cells in the grid.

**FALSE**

any crossing of the class variables that includes a missing value does not produce a row or columns of cells in the grid.

By default, missing class values are included in the classification. When the data contains missing classification values, cells are created for the missing classes. The classification headers for the missing values are blank for missing string values or a dot for missing numeric values. You can use the `INCLUDEMISSINGCLASS=FALSE` option to exclude the missing values. If you want to keep the missing values, then you can create a format that specifies more meaningful headings for the missing classes. For an example, see "Missing Class Values" in *SAS Graph Template Language: User's Guide*.

**Note:**

ODS Graphics does not support Unicode values in user-defined formats in *SAS 9.4M2* and in earlier releases. Starting with *SAS 9.4M3*, ODS Graphics supports Unicode values in user-defined formats only if they are preceded by the (*ESC*) escape sequence. Example: "(*ESC*){unicode beta}". ODS Graphics does not support an escape character that is defined in an ODS ESCAPECHAR statement in user-defined formats.

**Default**

TRUE

See "*boolean*" on page 1479 for other Boolean values that you can use.

**INSET=(variable-list)**

specifies what information is displayed in an inset. The `variable-list` defines one or more variables whose names and values appear as a small table in the data cells. The variables can be either numeric or character. Variable names are separated by spaces.
Requirement
No predefined information is available for the inset. You must create the desired inset information as part of your input data. See “Creating Your Inset Data” on page 71.

Note
The variable values are associated with the data cells by data order. That is, the first observation from all the variables in variable-list are used in the first data cell, the second observation from all variables in variable-list are used in the second data cell, and so on. If a value is missing for an observation, then the corresponding name-value pair is skipped in the affected data cell.

Tip
The location and appearance of the inset is controlled by the INSETOPTS= option.

See
“Adding Insets to Your Graph” in SAS Graph Template Language: User’s Guide

INSETOPTS=(appearance-options)
specifies location and appearance options for the inset information. The appearance options can be any one or more of the following values:

AUTOALIGN=NONE | AUTO | (location-list)
specifies whether the inset is automatically aligned within the layout.

NONE
does not automatically align the inset.

This inset’s position is set by the HALIGN= and VALIGN=appearance-options.

AUTO
attempts to center the inset in the area that is farthest from any surrounding markers.

Data cells might have different inset placements.

(location-list)
restricts the inset’s possible locations to those locations in the specified location-list, and uses the location-list position that least collides with the data cell’s other graphics features.

The location-list is a space-separated list that can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Example  AUTOALIGN=(TOPRIGHT TOPLEFT)

Default  NONE

Interaction
When AUTOALIGN=AUTO or (location-list), the HALIGN= and VALIGN= options are ignored.

BACKGROUNDCOLOR=style-reference | color
specifies the color of the inset background.

style-reference
specifies a style reference in the form style-element : style-attribute.

Only the COLOR and CONTRASTCOLOR style attributes are valid.

Default  The background is transparent. No color is assigned.
BORDER=TRUE | FALSE
specifies whether a border is displayed around the inset.

Default  FALSE

See  “boolean” on page 1479 for other Boolean values that you can use.

CONTENTDISPLAY=LABELVALUE | VALUE
specifies whether the variable information that is displayed in the inset includes the column label and value, or only the column value.

Note: This feature applies to SAS 9.4M2 and to later releases.

If a label is not assigned to a column, then the column name is used as the label for that column. Consider the following inset data:

```
Obs  _TYPE_  Value  Pr > F
1     SS1     94.359  <.0001
```

The following figure shows the effect that the CONTENTDISPLAY= option has on the content of an inset that displays this data.

<table>
<thead>
<tr>
<th>LABELVALUE</th>
<th>LABELVALUE with SEPARATOR=&quot;=&quot;</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>F Value 94.359</td>
<td>F Value = $4.359</td>
<td>94.359</td>
</tr>
<tr>
<td>Pr &gt; F &lt;.0001</td>
<td>Pr &gt; F = &lt;.0001</td>
<td>&lt;.0001</td>
</tr>
</tbody>
</table>

Default  LABELVALUE

Tip  Use the SEPARATOR= option to specify a separator other than the default blank space.

DATASCHEME=LIST | MATCHED
specifies the scheme that was used to merge the inset information into the analysis data.

LIST
one-to-one merging (no BY statement) was used to merge the inset and analysis data.

The variable values are associated with the cells of the data lattice by using data order. That is, the inset variable values in the first observation are used in the inset for the first cell, the inset variable values in the second observation are used in the inset for the second cell, and so on.

MATCHED
match-merging (using a BY statement) was used to merge the inset and analysis data.

Defaults  LIST if the data is a SAS data set

MATCHED if the data is a CAS in-memory table

Tip  MATCHED is the preferred data scheme for merging the inset and analysis data.
See “Adding Insets to Classification Panels” in SAS Graph Template Language: User’s Guide

**HALIGN=LEFT | CENTER | RIGHT**
specifies the horizontal alignment of the inset.

Default: LEFT

Interaction: This option has an effect only when this layout is nested within a REGION layout or when this layout is nested in an overlay-type layout and AUTOALIGN=None.

**OPAQUE=TRUE | FALSE**
specifies whether the inset background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When OPAQUE=FALSE, the background color is not used.

See “boolean” on page 1479 for other Boolean values that you can use.

**SEPARATOR="string"**
specifies a new separator for the column label and value.

Note: This feature applies to SAS 9.4M2 and to later releases.

Default: A blank space

Interaction: This option is ignored when CONTENTDISPLAY=VALUE.

**TEXTATTRS=style-element | style-element (text-options) | (text-options)**
specifies the text properties of the entire inset, excluding the title.

Default: The GraphDataText style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**TITLE="string"**
specifies a title for the inset.

The title is added at the top of the inset and spans the full inset width.

Note: Space is not reserved for the title when this value is not specified.

Tip: Text properties for the title string can be specified with TITLEATTRS=.

**TITLEATTRS=style-element | style-element (text-options) | (text-options)**
specifies the text properties of the inset’s title string.

Default: The GraphValueText style element.
VALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment of the inset.
Default TOP
Interaction This option has effect only when this layout is nested within a REGION layout or when this layout is nested in an overlay-type layout and AUTOALIGN=None.
Requirements The options must be enclosed in parentheses.
Each option must be specified as a name = value pair, and each pair must be separated by a space.

OPAQUE=TRUE | FALSE
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).
Default FALSE
Interaction When this option is set to FALSE, the background color is not used.
See “boolean ” on page 1479 for other Boolean values that you can use.

ORDER=ROWMAJOR | COLUMNMAJOR
specifies whether data cells are populated by column priority or by row priority.
ROWMAJOR
fills the data cells by rows, from the starting position.
COLUMNMAJOR
fills the data cells by columns, from the starting position.
Default ROWMAJOR
Interaction The starting point for rendering data cells is controlled by the START= option. See the START= option for examples.

OUTERPAD=AUTO | dimension | (pad-options)
specifies the amount of extra space to add outside the layout border.
AUTO
specifies that the default outside padding for this component is used.
dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.
(pad-options)
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.
The list can contain one or more of the following pad options:
LEFT=dimension
specifies the amount of extra space to add to the left side.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
“Text Options” on page 1525 for available text-options.
RIGHT=\textit{dimension} \\
\quad \text{specifies the amount of extra space to add to the right side.}

\textbf{TOP=\textit{dimension}} \\
\quad \text{specifies the amount of extra space to add to the top.}

\textbf{BOTTOM=\textit{dimension}} \\
\quad \text{specifies the amount of extra space to add to the bottom.}

\textbf{Note} \hspace{0.5cm} \text{Sides that are not assigned padding are padded with the default amount.}

\textbf{Tip} \hspace{0.5cm} \text{Use pad-options to create non-uniform padding.}

\textbf{Default} \hspace{0.5cm} \text{AUTO}

\textbf{Note} \hspace{0.5cm} \text{The default units for \textit{dimension} are pixels.}

\textbf{See} \hspace{0.5cm} \text{“\textit{dimension}” on page 1480}

\textbf{PAD=\textit{dimension} \textbar{} (pad-options)} \\
\quad \text{specifies the amount of extra space that is added inside the layout border.}

\textit{dimension} \\
\quad \text{specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.}

\textbf{(pad-options)} \\
\quad \text{a space-separated list of one or more name-value-pair pad options enclosed in parentheses.}

\textbf{Note} \hspace{0.5cm} \text{Sides that are not assigned padding are padded with the default amount.}

\textbf{Tip} \hspace{0.5cm} \text{Use pad-options to create non-uniform padding.}

\textbf{Note} \hspace{0.5cm} \text{The default units for \textit{dimension} are pixels.}
See “dimension” on page 1480

**PANELNUMBER=** *positive-integer*

specifies the number of the panel to produce. This option enables you to partition a large grid into a number of smaller sized grids under these conditions:

- You set a grid size explicitly (ROWS= and COLUMNS= options).
- The grid size (gridrows x gridcolumns) is smaller than the total number of classifier levels.
- You execute the template N times and increment the panel number each time. N is determined by CEIL\((\text{total-classification-levels} / \text{gridrows} \times \text{gridcolumns})\).

**Default** 1

**Example** Suppose there are two classifiers (CLASS1 has 10 unique values and CLASS2 has 11 unique values). By setting some smaller grid size, say ROWS=3 and COLUMNS=4, and making the value of PANELNUMBER= a dynamic or macro variable, you can create 10 panels (9 panels with 12 data cells and 1 panel with 2 data cells) that collectively display all 110 possible crossings. You simply invoke PROC SGRENDER or a DATA step 10 times, incrementing the dynamic value for PANELNUMBER each time.

**ROLENAMEn=(role-name-list)**

specifies user-defined roles for information contained in data columns. The role name list is a space-separated list role-name=column pairs.

**Default** no user-defined roles

**Requirement** The role names that you choose must be unique and different from the predefined roles.

**Example** The following example assigns the column Obs to the user-defined role TIP1.

```
ROLENAMEn=(TIP1=OBS)
```

**ROWAXISOPTS=**(axis-options)

specifies Y-axis options for all rows.

**Requirement** Axis options must be enclosed in parentheses and separated by spaces.

**See** “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” on page 1149 for a list of options.

**ROW2AXISOPTS=**(axis-options)

specifies Y2-axis options for all rows.

**Requirement** Axis options must be enclosed in parentheses and separated by spaces.

**Interaction** This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement’s YAXIS= option to map data to the Y2 axis, then this option is ignored. For
more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974

See “Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL” on page 1149 for a list of options.

**ROWDATARANGE=** AUTO | UNIONALL | UNION

specifies how the Y-axes of instances of the graph-prototype are scaled.

**AUTO**

- automatically selects the Y-axis scale based on the ROWWEIGHT= option and the column axis type.

  The Y-axis scale is selected as follows:

  - When ROWWEIGHT=EQUAL (default), UNIONALL is selected.
  - When ROWWEIGHT=PROPORTIONAL and the row axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

**UNIONALL**

- scales the Y-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

**UNION**

- scales the Y-axis data ranges separately for each row in the layout on a per-panel basis.

  The scaling does not span multiple panels.

Default AUTO

**Tip**

Use the ROWAXISOPTS= option to control shared axis features.

See The ROWWEIGHT= option.

The PANELNUMBER= option for information about how to create multiple panels.

**ROW2DATARANGE=** AUTO | UNIONALL | UNION

specifies how the Y2-axes of instances of the graph-prototype are scaled.

**AUTO**

- automatically selects the Y2-axis scale based on the ROWWEIGHT= option and the column axis type.

  The Y2-axis scale is selected as follows:

  - When ROWWEIGHT=EQUAL (default), UNIONALL is selected.
  - When ROWWEIGHT=PROPORTIONAL and the row axis is discrete, UNION is selected. Otherwise, UNIONALL is selected.

**UNIONALL**

- scales the Y2-axis data ranges across all layout rows and panels (when PANELNUMBER= is in effect).

**UNION**

- scales the Y2-axis data ranges separately for each row in the layout on a per-panel basis.

  The scaling does not span multiple panels.

Default AUTO
Interaction

This option is needed only if you use a plot statement that supports a secondary Y2 axis. If you do not use that statement’s YAXIS= option to map data to the Y2 axis, then this option is ignored. For more information about how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974.

Tip

Use the ROW2AXISOPTS= option to control shared axis features.

See

The ROWWEIGHT= option.

The PANELNUMBER= option for information about how to create multiple panels.

ROWGUTTER=\textit{dimension}

specifies the amount of empty space between the rows.

Default 0

Note If there are \( n \) rows, then there are \( n-1 \) gutters.

See “\textit{dimension}” on page 1480

ROWS=\textit{integer}

specifies the number of rows in the layout.

Defaults If this option is not specified and COLUMNS= is specified, then the number of data cells (and rows) increases dynamically to allow all classifier values to be presented.

If both this option and COLUMNS= are specified, then a grid of that size is created, regardless of the number of classifier values. If the number of classifier values is greater than the grid size, then no graphs are created for some classifier values. If the number of classifier values is small and the grid size large, then there might be empty cells created.

Interactions The overall grid size is constrained by the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. As the grid size grows, the cell size shrinks. To control the minimum size of a cell use the CELLHEIGHTMIN= and CELLWIDTHMIN= options.

The \textit{START=} and \textit{ORDER=} options affect how the rows are populated.

Tip The PANELNUMBER= option enables you to create multiple smaller grids that completely partition the classifier values.

ROWWEIGHT=\textsc{equal} | \textsc{proportional}

specifies how weights are assigned to the row heights.

\textsc{equal}

all rows have equal height.

\textsc{proportional}

each row has a height that is proportional to the number of discrete midpoint values that it contains.
Restriction

At least one row axis must be discrete in order for PROPORTIONAL to have any effect. Otherwise, EQUAL is used.

Interactions

When ROWDATARANGE=UNIONALL, PROPORTIONAL is ignored and EQUAL is used.

When PROPORTIONAL is in effect, ROWDATARANGE=AUTO is interpreted as UNION.

If all of the following conditions are true, then the discrete axis is used to proportion the rows: PROPORTIONAL is in effect, both the Y and Y2 axes are used, and only one of the two axes is discrete. When both axes are discrete, the Y axis is used to proportion the rows.

When PROPORTIONAL is in effect, the OFFSETMIN= and OFFSETMAX= axis options are ignored in ROWAXISOPTS= and ROW2AXISOPTS=. The axis offsets are always set to one-half of the midpoint spacing regardless of plot type.

Default

EQUAL

SHRINKFONTS=TRUE | FALSE

specifies whether fonts in the layout are scaled down.

Default

FALSE

Note

If SHRINKFONTS=TRUE, the font scaling applies to all child layouts within the parent layout. If a child layout also specifies SHRINKFONTS=TRUE, the fonts for the children of that child layout are further scaled down.

See

“boolean” on page 1479 for other Boolean values that you can use.

SKIPEMPTYCELLS=TRUE | FALSE

specifies whether the external axes skip the empty cells in a partially filled grid.

TRUE

skips empty cells and "snaps" the external axes to the nearest data cell, both vertically and horizontally.

Though the empty cells are not displayed, the data cells in the grid are not enlarged to fill the area.

FALSE

displays external axes at their normal locations, even if there are empty cells at one or more of the locations.

Whenever the total number of classifier crossings (data cells) is not evenly divisible by the panel size (columns * rows), the last panel is partially filled with data cells and padded with empty cells to complete the grid.

Here is an example of a data panel that consists of 16 data cells arranged in a 4-column, 3-row grid. The following figure shows the default appearance of the last panel:
When SKIPEMPTYCELLS=TRUE, the empty padding cells of the last panel are removed and external axis ticks and tick values snap to the data cells:

Note that SKIPEMPTYCELLS=TRUE removes only the empty padding cells on the last panel. It does not remove any data cells that have no crossing values and therefore no graph (these data cells are displayed when SPARSE=TRUE).

Default: FALSE

See “boolean” on page 1479 for other Boolean values that you can use.

**SORTORDER=(COLUMNVAR=sort-order-keyword | ROWVAR=sort-order-keyword)**

specifies the order of the cells along the columns and rows.

- **COLUMNVAR**
  - the column role.

- **ROWVAR**
  - the row variable role.
**sort-order-keyword**

a sort-order keyword.
The keyword can be one of the following:

**AUTO**
sorts using DATA for character data and ascending unformatted for numeric data.

**DATA**
retains the data order.

**Note**
In SAS 9.4M6 and SAS 9.4M5, when data stored in a CAS in-memory table is used and DATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify **ASCENDINGFORMATTED** or **DESCENDINGFORMATTED**.
Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA is specified, DATA is ignored, and AUTO is used instead.

**ASCENDINGFORMATTED**
sorts in ascending order, using the formatted values.

**DESCENDINGFORMATTED**
sorts in descending order, using the formatted values.

**Defaults**
AUTO for all roles if the data is a SAS data set

**ASCENDINGFORMATTED** for all roles if the data is a CAS in-memory table

**Tip**
The placement of the cells within the layout also depends on the starting location, which is controlled by the **START=** option.

**SPARSE=TRUE | FALSE**
specifies whether crossings of the class variables include only the crossings in the data or all possible crossings.

**FALSE**
specifies that data cells are created only for crossings of the class variables that are in the data.

**TRUE**
specifies that the number of data cells is the product of the unique values for each classification variable.

By default, if a crossing of the class variables has a missing value as part of the data, then a data cell is created for it.

Here is an example of a classification panel where the classification variables are COUNTRY and STATE. There are 3 distinct values of COUNTRY (Canada, Mexico, and U.S.A.) Within Canada and Mexico there are 4 states, and within U.S.A. there are 8 states. All state names are unique to each country. Therefore, there are 16 unique STATE values and 48 unique crossings of COUNTRY and STATE, but there are data for only 16 of the crossings.

Assume that a data panel layout is created with **COLUMNS=6** and **SPARSE=TRUE**, meaning to display all possible crossings. This is what the first row would look like. Blank data cells are added whenever there are no data values for a crossing:
When SPARSE=FALSE the crossings of the classifiers with no data are automatically removed. This compacts the display:

**SPARSE = TRUE**

<table>
<thead>
<tr>
<th>Canada</th>
<th>Canada</th>
<th>Canada</th>
<th>Canada</th>
<th>Canada</th>
<th>Canada</th>
</tr>
</thead>
<tbody>
<tr>
<td>British Columbia</td>
<td>Ontario</td>
<td>Quebec</td>
<td>Saskatchewan</td>
<td>Baja California Norte</td>
<td>Campeche</td>
</tr>
</tbody>
</table>

When SPARSE=FALSE the crossings of the classifiers with no data are automatically removed. This compacts the display:

**SPARSE = FALSE**

<table>
<thead>
<tr>
<th>Canada</th>
<th>Canada</th>
<th>Canada</th>
<th>Canada</th>
<th>Mexico</th>
<th>Mexico</th>
</tr>
</thead>
<tbody>
<tr>
<td>British Columbia</td>
<td>Ontario</td>
<td>Quebec</td>
<td>Saskatchewan</td>
<td>Baja California Norte</td>
<td>Campeche</td>
</tr>
</tbody>
</table>

Default **FALSE**

See “**boolean**” on page 1479 for other Boolean values that you can use.

**START=TOPLEFT | BOTTOMLEFT**

indicates whether to start populating the grid cells from the top left or bottom left corner. If ORDER=ROWMAJOR (the default) and START=TOPLEFT (the default), then a 2 row 2 column grid is populated as shown in the following figure.

```
+---------------+---------------+
|   Cell 1      |   Cell 2      |
+---------------+---------------+
|   Cell 3      |   Cell 4      |
+---------------+---------------+
```

If ORDER=ROWMAJOR (the default) and START=BOTTOMLEFT, then a 2 row 2 column grid is populated as shown in the following figure.

```
+---------------+---------------+
|   Cell 3      |   Cell 4      |
+---------------+---------------+
|   Cell 1      |   Cell 2      |
+---------------+---------------+
```

If ORDER=COLUMNMAJOR and START=BOTTOMLEFT, then a 2 row 2 column grid is populated as shown in the following figure.

```
+---------------+---------------+
|   Cell 3      |   Cell 4      |
+---------------+---------------+
|   Cell 1      |   Cell 2      |
+---------------+---------------+
If ORDER=COLUMNMAJOR and START=TOPLEFT, then a 2 row 2 column grid is populated as shown in the following figure.

**SIDEBAR Optional Argument**

**ALIGN=** BOTTOM | TOP | LEFT | RIGHT

specifies the sidebar's location within the layout. You can specify up to four SIDEBAR blocks in a LAYOUT DATAPANEL, one for each of the bottom, top, left, and right sidebar positions.

- The LAYOUT DATAPANEL automatically aligns a sidebar with the layout columns or rows.
- Only one statement (such as ENTRY or DISCRETELEGEND) or one layout block (such as LAYOUT GRIDDED) is allowed in a SIDEBAR block. To create multi-line text in a sidebar, nest ENTRY statements within a LAYOUT GRIDDED block.

**Default**  TOPLEFT

**SPACEFILL=** TRUE | FALSE

specifies whether to fill all the area of the sidebar with its contents.

**Default**  TRUE

**Tip**  To prevent a layout block within the sidebar from expanding to the sidebar boundaries, set this option to FALSE.

**See**  “boolean ” on page 1479 for other Boolean values that you can use.
Details

Statement Description

The LAYOUT DATAPANEL statement creates a grid of graphs, based on the values of one or more classifications variables. The main differences between this layout and the DATALATTICE layout are that this layout supports more than two classification variables, and it provides more control over the grid layout.

By default, the number of cells in the layout is determined by a crosstabulation table of all the classification variables plus any empty cells needed to complete the last row or column of the grid. The contents of each data cell are based on a graph prototype that you specify in the graph-prototype-block. You can enhance the display using one or more sidebar-statement-blocks. For classification variables that have many values, you can use the COLUMNS= option or the ROWS= option, or both with the PANELNUMBER= option to generate multiple panel displays.

The order of the value pairings for the classification variables is determined by the order that the variables are specified on the CLASSVARS= argument. The last named variable's values vary most rapidly (like nested DO loops). Variable values are always returned in data order.

By default, the first data cell to be filled is in the layout's top left corner, and data cells are filled from left-to-right, top-to-bottom. Use the START= option to change the starting data cell to the bottom left corner, and use the ORDER= option to determine whether data cells fill by column or by row. See the START= option for illustrations on how START= and ORDER= interact to manage the fill sequence for data cells.

Note: The DATAPANEL layout is designed to be the outermost layout in the template.

Prototype Block

You must specify a single graph-prototype-block within the LAYOUT DATAPANEL block, using the following syntax:

LAYOUT PROTOTYPE <options>;
   GTL-statements;
ENDLAYOUT;

The graph-prototype-block determines the graphical content of each data cell and is repeated within each data cell, based on the subsets of the classification variables.

For more information about the LAYOUT PROTOTYPE block and the list of available options, see "LAYOUT PROTOTYPE Statement" on page 173.

Sidebar Blocks

A LAYOUT DATAPANEL enables you to display sidebars outside of the axis areas. A sidebar spans across columns or rows and is useful for displaying information that applies to all of the columns or all of the rows. For example, sidebars are useful for displaying a legend.

A SIDEBAR statement has the following syntax:

SIDEBAR <options>;
   GTL-statement(s);
ENDSIDEBAR;
The following example shows a SIDEBAR block that displays a legend at the top of the layout grid.

```plaintext
sidebar / align=top;
  discretelegend 'p' 'a' / across=2;
endsidebar;
```

Creating Your Inset Data

When you use the **INSET** option to insert an inset, no predefined information is available for the inset. You must create the desired inset information as part of your input data. This is most typically done as follows (see the chapter on classification panels and the chapter on insets in *SAS Graph Template Language: User’s Guide* for complete examples):

- Create a separate data set for the inset columns making sure that the column names are different from the other columns used in graph. The number of observations of inset data should match the number of cells in the classification panel. The ordering of the inset observations should be the same as population order of the cells of the classification panel, taking into account the `CLASSVARS=` argument and the `ORDER=` and `START=` options. Typically, the number of observations for the inset data is smaller than the other input data for the graph.
- Merge the inset data set with the data set for the graph using a DATA or PROC SQL step. Do not match-merge the observations of the two data sets (no BY processing). The resulting data set typically has the inset columns padded with missing values.
- Use the merged data set to produce the graph, specifying the inset column names in this option’s `variable-list`.

Example: LAYOUTDATAPANEL Statement

This example shows a four-column, three-row data panel using two classification variables. With this layout, each data cell is subsetted and labeled with the values of the classification variables.

- The `ROWDATARANGE=UNION` option assures that an axis range is computed separately for each row using the data ranges of all Y= columns in that row. This facilitates the visual comparison of the data cells.
- A SIDEBAR block is used to place the legend at the bottom of the lattice.

Example Graph

The following graph was generated by the “Example Program” on page 100:
Example Program

```plaintext
proc template;
  define statgraph layoutdatapanel;
  begingraph;
    entrytitle "Annual Furniture Sales Comparisons";
    layout datapixel classvarg=(country year) /
      columns=4 rows=3 rowdatarange=union
      headerlabeldisplay=value
      headerbackgroundcolor=GraphAltBlock:color
      rowaxisopts=(display=(tickvalues) griddisplay=on
        linearopts=(tickvalueformat=dollar12.))
      columnaxisopts=(display=(tickvalues)
        timeopts=(tickvalueformat=monname3.));
    layout prototype / cycleattrs=true;
      seriesplot x=month y=TotalActual / name="Actual";
      seriesplot x=month y=TotalPredict / name="Predict";
    endlayout;
    sidebar / align=top;
      discretelegend "Actual" "Predict" / border=false;
    endsidebar;
  endlayout;
  endgraph;
  end;
run;

proc summary data=sashelp.prdsal2 nway;
  class country year month;
  var actual predict;
```
output out=prdsal2 sum=TotalActual TotalPredict;
run;

proc sgrender data=prdsal2 template=layoutdatapanel;
run;

LAYOUT GLOBALLEGEND Statement

Creates a compound legend containing multiple discrete legends positioned at the bottom of a graph.

Restrictions:
Only one global legend is allowed in a graph.
Continuous legends are not supported inside the global legend.

Requirements:
You must place the GLOBALLEGEND block directly inside the BEGINGRAPH block. Do not embed it in any other GTL statement block.
When the LAYOUT GLOBALLEGEND block is used, all of the template's legend statements must be specified within the LAYOUT GLOBALLEGEND block. Any legend statements that are specified outside of the LAYOUT GLOBALLEGEND block are ignored.

See:
“DISCRETELEGEND, MERGEDLEGEND Statements” on page 1234

Syntax

LAYOUT GLOBALLEGEND <options>;
<discreteLegend-statement(s) | mergedLegend-statement(s)>;
ENDLAYOUT;

Summary of Optional Arguments

Appearance options

BORDER=TRUE | FALSE
specifies whether a border is drawn around the layout.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the border line around the layout.

DISPLAYCLIPPED=TRUE | FALSE
specifies whether the global legend is displayed when any portion of its nested legends cannot be fully rendered because of space constraints.

GUTTER=dimension
specifies the gap between nested layouts.

OUTERPAD=AUTO | dimension | (pad-options)
specifies the amount of extra space to add outside the layout border.

PAD=dimension | (pad-options)
specifies the amount of extra space that is added inside the layout border.

TYPE=ROW | COLUMN
specifies whether nested legends are arranged into a single row or column.

WEIGHTS=UNIFORM | PREFERRED | (weight-list)
specifies the preferred space allocation for the nested legends.
Legend title options

- **LEGENDTITLEPOSITION=LEFT | TOP**
  specifies the position of each nested legend’s title.

- **TITLE="string"**
  specifies a title for the global legend.

- **TITLEATTRS=style-element | style-element (text-options) | (text-options)**
  specifies the color and font attributes of the global legend title.

Location options

- **HALIGN=CENTER | LEFT | RIGHT**
  specifies the layout’s horizontal alignment within the graph area that is defined by the BEGINGRAPH block.

Optional Arguments

- **BORDER=TRUE | FALSE**
  specifies whether a border is drawn around the layout.

  Default: FALSE

  Interaction: If this option is set to FALSE, then the BORDERATTRS= option is ignored.

  See: “boolean “ on page 1479 for other Boolean values that you can use.

- **BORDERATTRS=style-element | style-element (line-options) | (line-options)**
  specifies the attributes of the border line around the layout.

  Default: The GraphBorderLines style element.

  Interaction: BORDER= TRUE must be set for this option to have any effect.

  See: “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

  “Line Options” on page 1522 for available line-options.

- **DISPLAYCLIPPED=TRUE | FALSE**
  specifies whether the global legend is displayed when any portion of its nested legends cannot be fully rendered because of space constraints. When the graph size is reduced, parts of a nested legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and this option is FALSE, the entire global legend is removed from the graph and the space for it is reclaimed by the remainder of the graph. When this option is TRUE, the global legend always appears, even if some parts of the nested legends have been clipped.

  Default: FALSE

  Interaction: This option overrides any DISPLAYCLIPPED option that is set on its nested legend statements.

  See: “boolean “ on page 1479 for other Boolean values that you can use.

- **GUTTER=dimension**
  specifies the gap between nested layouts.

  Default: 0
Note: The default units for dimension are pixels.

See: "dimension" on page 1480

HALIGN=CENTER | LEFT | RIGHT
specifies the layout’s horizontal alignment within the graph area that is defined by the BEGINGRAPH block.

Default: CENTER

Note: When CENTER is in effect and the outermost layout is an overlay-type layout, the global legend is centered below the wall area if it can fit within the wall width.

LEGENDTITLEPOSITION=LEFT | TOP
specifies the position of each nested legend’s title. Specifying LEFT places each title to the left of the legend items for that legend. Specifying TOP places each title above the legend items for that legend.

Default: LEFT

OUTERPAD=AUTO | dimension | (pad-options)
specifies the amount of extra space to add outside the layout border.

AUTO
specifies that the default outside padding for this component is used.

dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

(pad-options)
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can contain one or more of the following pad options:

LEFT=dimension
specifies the amount of extra space to add to the left side.

RIGHT=dimension
specifies the amount of extra space to add to the right side.

TOP=dimension
specifies the amount of extra space to add to the top.

BOTTOM=dimension
specifies the amount of extra space to add to the bottom.

Note: Sides that are not assigned padding are padded with the default amount.

Tip: Use pad-options to create non-uniform padding.

Default: AUTO

Notes: The default units for dimension are pixels.
Starting with SAS 9.4M1, the default padding between the global legend and the plot area (including the axes) is increased to 10 pixels. If the new default padding is not desirable, then use this option to adjust it.

**PAD=** `dimension | (pad-options)`  
specifies the amount of extra space that is added inside the layout border.

- **dimension**  
specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

- **(pad-options)**  
a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

- **LEFT=** `dimension`  
specifies the amount of extra space added to the left side.  
  Default: 0

- **RIGHT=** `dimension`  
specifies the amount of extra space added to the right side.  
  Default: 0

- **TOP=** `dimension`  
specifies the amount of extra space added to the top.  
  Default: 0

- **BOTTOM=** `dimension`  
specifies the amount of extra space added to the bottom.  
  Default: 0

**Note**  
Sides that are not assigned padding are padded with the default amount.

**Tip**  
Use `pad-options` to create non-uniform padding.

**Note**  
The default units for `dimension` are pixels.

**TITLE=** `"string"`  
specifies a title for the global legend.

- **Default** No title is displayed for the global legend.

- **Restriction** The string must be enclosed in quotation marks.

- **Tip** The title for the global legend is independent of the titles for its nested legends.
**TITLEATTRS=** *style-element | style-element (text-options) | (text-options)*  
specifies the color and font attributes of the global legend title.

**Default**  
The GraphLabelText style element.

**Interaction**  
For this option to have any effect, the **TITLE=** option must also be specified.

**See**  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element.*

“Text Options” on page 1525 for available *text-options.*

**TYPE=ROW | COLUMN**  
specifies whether nested legends are arranged into a single row or column.

**Default**  
ROW

**Interaction**  
When this option is set to ROW, the relative width of each legend is determined by the setting for the **WEIGHTS=** option.

**WEIGHTS=** *UNIFORM | PREFERRED | (weight-list)*  
specifies the preferred space allocation for the nested legends.

**UNIFORM**  
allocates an equal amount of space for all nested legends.

**PREFERRED**  
allocates the preferred amount of space for each nested legend.

**(weight-list)**  
a space-separated list of preferred space allocations, enclosed in parentheses. The list can combine numbers with the keyword PREFERRED. Each number is a proportional weight for the corresponding nested legend (the weights do not have to sum to 1.0). Keyword PREFERRED specifies that the preferred space should be allocated to the corresponding nested legend. The order of the weights that are specified in the list should correspond to the order of the legend statements that are nested in the GLOBALLEGEND layout.

**Default**  
UNIFORM

**Restriction**  
The option is supported only for **TYPE=ROW.**

**Tip**  
When a *weight-list* is specified, all the legends using PREFERRED get their preferred space. Any remaining space is divided among the legends, in proportion to the numeric values specified in the *weight-list.*

**Details**

A global legend layout can contain multiple discrete or merged legends. Continuous legends are not supported inside the global legend block.

A global legend is placed at the bottom of the graph, just above the footnote(s). All of the discrete or merged legend statements that are nested within the global legend block are arranged into a single row or column, depending on the setting for the **TYPE=** option.
Depending on the outermost layout type and the legend content, the legend is centered on the graph wall area or on the graph output area. For example, if the outermost layout is an OVERLAY layout, then when positioning the legend, the GLOBALLEGEND statement first attempts to center the legend on the graph wall area. If that position causes the legend to be clipped, then it attempts to center the legend on the entire output area instead. In that case, the legend might appear to be slightly off-center with respect to the graph.

Only one global legend block is permitted in a graph. The block must be located within the BEGINGRAPH block, but outside of the outermost layout block.

When a global legend block is used, only the legend statements that are specified within the global legend block are displayed in the graph. Any legend statements that are specified outside of the global legend block are ignored.

Example: LAYOUT GLOBALLEGEND Statement

The following graph was generated by the “Example Program” on page 106:

---

Example Program

```<proc template>
   define statgraph globallegend;
   begingraph;
      entrytitle "Prediction Ellipses";
      layout overlay;
      scatterplot x=petallength y=petalwidth / group=species name="sp";
      ellipse x=petallength y=petalwidth / type=predicted alpha=0.2
         name="p80" legendlabel="80%" outlineattrs=graphconfidence;
      ellipse x=petallength y=petalwidth / type=predicted alpha=0.05
         name="p95" legendlabel="95%" outlineattrs=graphconfidence2;
```

---
LAYOUT GRIDDED Statement

Assembles the results of nested GTL-statements into a grid.

Note: Empty cells might be omitted from the grid. See “Details” on page 114.

Syntax

LAYOUT GRIDDED <options>;
   GTL-statements;
ENDLAYOUT;

Summary of Optional Arguments

Appearance options

   BACKGROUNDCOLOR=style-reference | color
      specifies the color of the layout background.

   BORDER=TRUE | FALSE
      specifies whether a border is drawn around the layout.

   BORDERATTRS=style-element | style-element (line-options) | (line-options)
      specifies the attributes of the border line around the layout.

   COLUMNGUTTER=dimension
      specifies the amount of empty space between the columns.

   OPAQUE=TRUE | FALSE
      specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

   OUTERPAD=AUTO | dimension | (pad-options)
      specifies the amount of extra space to add outside the layout border.

   PAD=dimension | (pad-options)
      specifies the amount of extra space that is added inside the layout border.

   SHRINKFONTS=TRUE | FALSE
      specifies whether fonts in the layout are scaled down.

Grid options

   COLUMNS=integer
      specifies the number of columns in the layout.

   ROWS=integer

   proc sgrender data=sashelp.iris template=globallegend;
   run;

   layout globalLegend / type=column title="Sample Global Legend";
       discretelegend "sp" / title="Species:";
       discretelegend "p80" "p95" / title="Predictions:";
   endLayout;
   endgraph;
   end;
   run;

   proc sgrender data=sashelp.iris template=globallegend;
   run;
specifies the number of rows in the layout.

**Layout options**

```
ROWGUTTER=dimension
```

specifies the amount of empty space between the rows.

**Legend options**

```
LOCATION=INSIDE | OUTSIDE
```

specifies whether the legend appears inside or outside the plot area when nested within an overlay-type layout.

**Location options**

```
AUTOALIGN=NULL | AUTO | (location-list)
```

specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout.

```
HALIGN=CENTER | LEFT | RIGHT | number
```

specifies this layout's horizontal alignment within its parent when nested within an overlay-type or REGION layout.

```
VALIGN=CENTER | TOP | BOTTOM | number
```

specifies this layout's vertical alignment within its parent when nested within an overlay-type or REGION layout.

**Panel options**

```
ORDER=ROWMAJOR | COLUMNMAJOR
```

specifies whether data cells are populated by column priority or by row priority.

**Optional Arguments**

```
AUTOALIGN=NULL | AUTO | (location-list)
```

specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout.

```
NONE
```

does not automatically align the layout within its overlay-type parent layout.

This layout's position within its parent layout is therefore set by the HALIGN= and VALIGN= options.

```
AUTO
```

within the overlay-type parent layout, attempts to center the layout in the area that is farthest from any surrounding data point markers.

This option is available only if the parent layout contains a scatter plot. Otherwise, it is ignored.

```
(location-list)
```

within the parent layout, restricts the layout's possible locations to those locations in the specified location-list, and uses the location-list position that least collides with the parent layout's other graphics features.

The location-list is a space-separated list that can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

**Default**

```
NONE
```
### RESTRICTION
This option is ignored if this layout statement is the outermost layout or if the parent layout is not an overlay-type layout.

### INTERACTIONS
When this option is not NONE and the parent layout is an overlay-type layout, the **HALIGN**= and **VALIGN**= options are ignored.

This option is ignored if **LOCATION**= OUTSIDE.

See "LAYOUT OVERLAY Statement" on page 149 for more information about how child positions are determined in an overlay-type layout.

### BACKGROUND COLOR
**BACKGROUNDCOLOR=**<style-reference> | <color>
specifies the color of the layout background.

- **style-reference**
  specifies a style reference in the form **style-element**:**style-attribute**.
  Only the style attribute named COLOR or CONTRASTCOLOR is used.

- **color**
  specifies a color by name.

See color on page 1480

**Default** The GraphBackground:Color style reference.

**Interaction** **OPAQUE**=TRUE must be in effect in order for the color to be seen. By default, OPAQUE=FALSE.

**BORDER**=**TRUE** | **FALSE**
specifies whether a border is drawn around the layout.

**Default** FALSE

**Interaction** If this option is set to FALSE, then the BORDERATTRS= option is ignored.

See "boolean " on page 1479 for other Boolean values that you can use.

**BORDERATTRS=**<style-element> | <style-element> (<line-options>) | (<line-options>)
specifies the attributes of the border line around the layout.

**Default** The GraphBorderLines style element.

**Interaction** **BORDER**= TRUE must be set for this option to have any effect.

See "General Syntax for Attribute Options" on page 1519 for the syntax on using a **style-element**.

"Line Options" on page 1522 for available **line-options**.

**COLUMN GUTTER=**<dimension>
specifies the amount of empty space between the columns.

**Default** 0

**Note** If there are *n* columns, then there are *n*-1 gutters.

See "**dimension**" on page 1480
**COLUMNS=integer**
specifies the number of columns in the layout. This option is used to create a grid with a fixed number of columns, without concern for how many rows. For example, the following settings ensure that columns 1 and 2 in the first row are filled with content, as shown in the figure:

```
layout gridded / columns=2 order=rowmajor border=true ;
   entry '1' /border=true;
   entry '2' /border=true;
   entry '3' /border=true;
endlayout;
```

**Defaults**
If ORDER= ROWMAJOR, then the default is 1.

If ORDER=COLUMNMAJOR, then as many columns are created as needed to satisfy the ROWS= request.

**Restriction**
Assuming ORDER=ROWMAJOR, if COLUMNS=n and there are m cells defined, and n > m, then only m columns are created (there are n - m cells with zero size).

**HALIGN=CENTER | LEFT | RIGHT | number**
specifies this layout’s horizontal alignment within its parent when nested within an overlay-type or REGION layout.

*number*
specifies the horizontal alignment as a fraction of the parent container’s width.

**Range**
0 to 1, where 0 is all the way to the left and 1 is all the way to the right.

**Interaction**
For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

**Default**
CENTER

**Restriction**
This option has effect only when this layout is nested within a REGION layout, or when this layout is nested in an overlay-type layout and AUTOALIGN=NONE.

**See**
“LAYOUT OVERLAY Statement” on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

**LOCATION=INSIDE | OUTSIDE**
specifies whether the legend appears inside or outside the plot area when nested within an overlay-type layout.

**Default**
INSIDE
Restriction
This option has effect only when the GRIDDED layout block appears within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see “LAYOUT OVERLAY Statement” on page 149.

Interactions
If this option is set to OUTSIDE, then the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The number setting for the alignment is invalid when the layout is positioned outside of the plot area.

The actual position is determined by this option’s setting plus the settings for the AUTOALIGN= or HALIGN= and VALIGN= options.

**OPAQUE=TRUE | FALSE**
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default FALSE

Interaction When this option is set to FALSE, the background color is not used.

See “boolean ” on page 1479 for other Boolean values that you can use.

**ORDER=ROWMAJOR | COLUMNMAJOR**
specifies whether data cells are populated by column priority or by row priority.

**ROWMAJOR**
fills all the columns in a row, from left to right, before going to the next row.

**COLUMNMAJOR**
fills all the rows in a column, from top to bottom, before going to the next column.

Default ROWMAJOR

Requirements If this option is set to COLUMNMAJOR, then the ROWS= option must be specified to indicate how many rows to fill before wrapping to the next column. The default number of rows is 1.

If this option is set to ROWMAJOR, then the COLUMNS= option must be specified to indicate how many columns to fill before wrapping to the next column. The default number of columns is 1.

Interactions The ROWS= option is ignored when ORDER=ROWMAJOR.

The COLUMNS= option is ignored when ORDER=COLUMNMAJOR.

See ROWS= on page 113

COLUMNS= on page 110

**OUTERPAD=AUTO | dimension | (pad-options)**
specifies the amount of extra space to add outside the layout border.

**AUTO**
specifies that the default outside padding for this component is used.
**dimension**

specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

**(pad-options)**

a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can contain one or more of the following pad options:

**LEFT=dimension**

specifies the amount of extra space to add to the left side.

**RIGHT=dimension**

specifies the amount of extra space to add to the right side.

**TOP=dimension**

specifies the amount of extra space to add to the top.

**BOTTOM=dimension**

specifies the amount of extra space to add to the bottom.

---

**Note**

Sides that are not assigned padding are padded with the default amount.

**Tip**

Use *pad-options* to create non-uniform padding.

---

**Default**  

**AUTO**

**Note**

The default units for *dimension* are pixels.

**See**

“*dimension*” on page 1480

**PAD=dimension | (pad-options)**

specifies the amount of extra space that is added inside the layout border.

**dimension**

specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

**(pad-options)**

a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

**LEFT=dimension**

specifies the amount of extra space added to the left side.

**Default**  

0

**RIGHT=dimension**

specifies the amount of extra space added to the right side.

**Default**  

0

**TOP=dimension**

specifies the amount of extra space added to the top.

**Default**  

0
BOTTOM=dimension
specifies the amount of extra space added to the bottom.

Default  0

Note  Sides that are not assigned padding are padded with the default amount.

Tip  Use pad-options to create non-uniform padding.

Note  The default units for dimension are pixels.

See  "dimension" on page 1480

ROWGUTTER=dimension
specifies the amount of empty space between the rows.

Default  0

Note  If there are n rows, then there are n-1 gutters.

See  "dimension" on page 1480

ROWS=integer
specifies the number of rows in the layout. This option is used to create a grid with a fixed number of rows, without concern for how many columns. For example, the following settings ensure that rows 1 and 2 in the first column are filled with content, as shown in the figure:

```
layout gridded / rows=2 order=columnmajor border=true ;
  entry '1' /border=true;
  entry '2' /border=true;
  entry '3' /border=true;
endlayout;
```

Defaults  If ORDER= COLUMNMAJOR, then the default is 1.

If ORDER=ROWMAJOR, then this option is ignored and as many rows are created as needed to satisfy the COLUMNS= request.

Restriction  Assuming ORDER=COLUMNMAJOR, if ROWS=n and there are m cells defined, and n > m, then only m rows are created (there are n - m cells with zero size).

SHRINKFONTS=TRUE | FALSE
specifies whether fonts in the layout are scaled down.

Default  FALSE

Note  If SHRINKFONTS=TRUE, the font scaling applies to all child layouts within the parent layout. If a child layout also specifies
SHRINKFONTS=TRUE, the fonts for the children of that child layout are further scaled down.

See “boolean” on page 1479 for other Boolean values that you can use.

**VALIGN=** CENTER | TOP | BOTTOM | number

specifies this layout’s vertical alignment within its parent when nested within an overlay-type or REGION layout.

number specifies the vertical alignment as a fraction of the parent container’s height.

Range 0 to 1, where 0 is on the bottom and 1 is on the top.

Interaction For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

Default CENTER

Restriction This option has effect only when this layout is nested within a REGION layout, or when this layout is nested in an overlay-type layout and AUTOALIGN=NONE.

See “LAYOUT OVERLAY Statement” on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

Details

A GRIDDED layout is commonly used to create small tables of text that are nested within other layouts. The layout might also be used to span and center a single entry (a legend, for example) across a set of grids. Or it might be used to display a grid of graphs when there is no need to scale axis data ranges or align graphs across grid cells.

The GRIDDED layout automatically decides how much area to allocate to cell contents:

- Text items have a fixed size based on the amount of text and the font properties.
- Graphs take up the remaining space.

The layout’s grid size is determined by the COLUMNS= and ROWS= options. The resulting columns and rows can be separated by areas called “gutters,” which are controlled by the COLUMNGUTTER= and ROWGUTTER= options.

By default, the results of the GTL statements are placed into the grid sequentially from left to right, wrapping to a new row each time the current row is filled. You can use the ORDER= option to fill cells from top to bottom down a column. In that case, the layout cells wrap to a new column each time the current column is filled. The GTL statements can include text statements, plot statements, and layout blocks. Each statement or layout block provides content for one cell in the grid.

If a statement or layout block for a grid cell does not produce any output, then the space for that cell might not be retained as an empty cell in the grid. In that case, the empty cell is removed, and the remaining cells (if any) fill the gap in the grid. A statement produces no output when required data for that statement does not
resolve. A layout block produces no output when it contains no statements or when none of the statements contained in the block produce any output.

Examples:

Example 1: Creating a 2-By-2 Grid of Plots

This example shows you how to create a simple 2-by-2 grid of plots. Each plot in the grid must be the same size and have independent axes. The GRIDDED layout is the ideal choice for these requirements. The following graph was generated by the “Example Program” on page 115:

Example Program

/* Define the graph template */
proc template;
  define statgraph graphgridded;
    begingraph / subpixel=on;
      entrytitle "Automotive Data";
      layout gridded / columns=2 rows=2 rowgutter=5 columngutter=5;
        /* Define graph for row 1 cell 1 */
        layout overlay / border=on pad=5
          xaxisopts=(label="Mean Highway MPG" griddisplay=on gridattrs=(color=lightgray pattern=dot))
          yaxisopts=(label="Type" labelposition=top);
          barchart x=type y=Mpg_Highway / stat=mean orient=horizontal
            barlabel=true barlabelformat=5.0;
          /* Define graph for row 1 cell 2 */
          layout overlay / border=on pad=5
            xaxisopts=(label="Mean Highway MPG" griddisplay=on gridattrs=(color=lightgray pattern=dot))
            yaxisopts=(label="Origin" labelposition=top);
            barchart x=origin y=Models_Produced / stat=mean orient=horizontal
              barlabel=true barlabelformat=5.0;
          /* Define graph for row 2 cell 1 */
          layout overlay / border=on pad=5
            xaxisopts=(label="Curb Weight (lbs)" griddisplay=on gridattrs=(color=lightgray pattern=dot))
            yaxisopts=(label="Type" labelposition=top);
            scatterplot x=cw y=hp / styleattrs=(color=blue) legend=""></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br"></br">"
Example 2: Creating an Inset

The following graph was generated by the “Example Program” on page 117:
Example Program

The GRIDDED layout offers the best way to nest a table of information inside another layout. In the GRIDDED layout, you can control the content, text justification, and fonts of columns. Because this example nests the GRIDDED layout within an OVERLAY layout, you can control where it appears within the plot area. The AUTOALIGN= option enables you to specify a prioritized list of possible positions where the layout should be drawn. The position actually used is the first one that avoids collision with the histogram. Also, the GRIDDED layout is set to be opaque so that the grid lines do not show through. This example also illustrates a reusable template in the sense that it works for any numeric column specified by the dynamic variable VAR. Also, SGE functions for computing the N, MEAN, STDDEV of the column are used in the table to compute the statistics as the template is executed.

```
proc template;
  define statgraph inset;
  dynamic VAR;
  begingroup;
    entrytitle "Distribution of " VAR;
    layout overlay / yaxisopts=(griddisplay=on);
    histogram VAR / scale=percent;
    layout gridded / columns=2
      autoalign=(topleft topright) border=true
        opaque=true backgroundcolor=GraphWalls:color;
      entry halign=left "N";
      entry halign=left eval(strip(put(n(VAR),12.0)));
      entry halign=left "Mean";
      entry halign=left eval(strip(put(mean(VAR),12.2)));
      entry halign=left "Std Dev";
      entry halign=left eval(strip(put(stddev(VAR),12.2)));
  endgroup;
end;```

Example 2: Creating an Inset
LAYOUT LATTICE Statement

Creates a grid of graphs that automatically aligns plot areas and tick display areas across grid cells to facilitate data comparisons among graphs.

Note: Empty cells might be omitted from the lattice. See “Cell Contents” on page 134.

Syntax

```
LAYOUT LATTICE <options>;
   GTL-statement(s) | cell-statement-block(s);
   <COLUMNAXES;
      COLUMNAXIS / axis-options;
      <... more-COLUMNAXIS-statements ...>
   ENDCOLUMNAXES;>
   <COLUMN2AXES;
      COLUMNAXIS / axis-options;
      <... more-COLUMNAXIS-statements ...>
   ENDCOLUMN2AXES;>
   <ROWAXES;
      ROWAXIS / axis-options;
      <... more-ROWAXIS-statements ...>
   ENDROWAXES;>
   <ROW2AXES;
      ROWAXIS / axis-options;
      <... more-ROWAXIS-statements ...>
   ENDROW2AXES;>
   <COLUMNHEADERS;
      GTL-statement(s);
   ENDCOLUMNHEADERS;
   <... more-header-statement-block(s) ...> >
   <SIDEBAR <options>;
      GTL-statement(s);
   ENDSIDEBAR;>
   <... more-sidebar-statement-blocks ...> >
ENDLAYOUT;
```
Summary of Optional Arguments

Appearance options

- **BACKGROUND**\(\text{COLOR}=\text{style-reference} | \text{color}\)
  - specifies the color of the layout background.

- **BORDER**=TRUE | FALSE
  - specifies whether a border is drawn around the layout.

- **BORDERATTRS**=style-element | style-element \(\text{(line-options)}\) | (line-options)
  - specifies the attributes of the border line around the layout.

- **COLUMNGUTTER**=dimension
  - specifies the amount of empty space that is between the columns.

- **OPAQUE**=TRUE | FALSE
  - specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

- **OUTERPAD**=AUTO | dimension | (pad-options)
  - specifies the amount of extra space to add outside the layout border.

- **PAD**=dimension | (pad-options)
  - specifies the amount of extra space that is added inside the layout border.

- **SHRINKFONTS**=TRUE | FALSE
  - specifies whether fonts in the layout are scaled down.

Cell options

- **COLUMNWEIGHTS**=UNIFORM | PREFERRED | \(\text{(weight-list)}\)
  - specifies the proportional width of each column relative to the overall grid width, not including the headings and sidebars.

- **ROWWEIGHTS**=UNIFORM | PREFERRED | \(\text{(weight-list)}\)
  - specifies the proportional height of each row relative to the overall grid height, not including the headings and sidebars.

Column options

- **COLUMN2DATARANGE**=DATA | UNION | UNIONALL
  - specifies how the X2-axis data ranges of graphs within the lattice columns are scaled.

- **COLUMNDATARANGE**=DATA | UNION | UNIONALL
  - specifies how the X-axis data ranges of graphs within the lattice columns are scaled.

Lattice options

- **COLUMNS**=integer
  - specifies the number of columns in the lattice.

- **ROWS**=integer
  - specifies the number of rows in the lattice.

- **SKIPEMPTYCELLS**=TRUE | FALSE
  - specifies whether row and column axes skip the unused cells in a partially filled lattice.

Layout options

- **ROWGUTTER**=dimension
  - specifies the amount of empty space between the rows.

Location options

- **AUTOALIGN**=NONE | AUTO | \(\text{(location-list)}\)
specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout.

\textbf{HALIGN=} \texttt{CENTER | LEFT | RIGHT | number}  
 specifies this layout's horizontal alignment within its parent when nested within an overlay-type or REGION layout.

\textbf{VALIGN=} \texttt{CENTER | TOP | BOTTOM | number}  
 specifies this layout's vertical alignment within its parent when nested within an overlay-type or REGION layout.

\textbf{Panel options}

\textbf{ORDER=} \texttt{ROWMAJOR | COLUMNMAJOR}  
 specifies whether data cells are populated by column priority or by row priority.

\textbf{Row options}

\textbf{ROW2DATARANGE=} \texttt{DATA | UNION | UNIONALL}  
 specifies how the Y2-axis data ranges of graphs within the lattice rows are scaled.

\textbf{ROWDATARANGE=} \texttt{DATA | UNION | UNIONALL}  
 specifies how the Y-axis data ranges of graphs within the lattice rows are scaled.

\textbf{Optional Arguments}

\textbf{AUTOALIGN=} \texttt{NONE | AUTO | (location-list)}  
 specifies whether this layout is automatically aligned within its parent when nested within an overlay-type layout.

\textbf{NONE}  
 does not automatically align the layout within its overlay-type parent layout.  
 This layout's position within its parent layout is therefore set by the \texttt{HALIGN=} and \texttt{VALIGN=} options.

\textbf{AUTO}  
 within the overlay-type parent layout, attempts to center the layout in the area that is farthest from any surrounding data point markers.

This option is available only if the parent layout contains a scatter plot. Otherwise, it is ignored.

\textbf{(location-list)}  
 within the parent layout, restricts the layout's possible locations to those locations in the specified \texttt{location-list}, and uses the \texttt{location-list} position that least collides with the parent layout's other graphics features.

The \texttt{location-list} is a space-separated list that can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

\textbf{Default} \texttt{NONE}

\textbf{Restriction}  
 This option is ignored if this layout statement is the outermost layout or if the parent layout is not an overlay-type layout.

\textbf{Interaction}  
 When this option is not NONE and the parent layout is an overlay-type layout, the \texttt{HALIGN=} and \texttt{VALIGN=} options are ignored.
See “LAYOUT OVERLAY Statement” on page 149 for more information about how child positions are determined in an overlay-type layout.

Example

In the following example, the LATTICE layout is the child of the OVERLAY layout. The child layout appears in either the top right or top left position, based on which position has more “unoccupied” area.

```plaintext
dynamic VAR STAT1 STAT2 STAT3;
layout overlay;
  histogram VAR;
  layout lattice / AUTOALIGN=(TOPRIGHT TOPLEFT)
  columns=1;
    entry STAT1;
    entry STAT2;
    entry STAT3;
  endlayout;
endlayout;
```

**BACKGROUND COLOR**=

```plaintext
STYLE-REFERENCE | COLOR
```

specifies the color of the layout background.

**STYLE-REFERENCE**

specifies a style reference in the form **STYLE-ELEMENT**: **STYLE-ATTRIBUTE**.

Only the style attribute named COLOR or CONTRASTCOLOR is used.

color

specifies a color by name.

See **color** on page 1480

**Default**
The GraphBackground:Color style reference.

**Interaction**

**OPAQUE**=TRUE must be in effect in order for the color to be seen. By default, OPAQUE=FALSE.

**BORDER**=TRUE | FALSE

specifies whether a border is drawn around the layout.

**Default**
FALSE

**Interaction**

If this option is set to FALSE, then the BORDERATRBS= option is ignored.

See “**boolean**” on page 1479 for other Boolean values that you can use.

**BORDERATRBS**=STYLE-ELEMENT | STYLE-ELEMENT (LINE-OPTIONS) | (LINE-OPTIONS)

specifies the attributes of the border line around the layout.

**Default**
The GraphBorderLines style element.

**Interaction**

**BORDER**= TRUE must be set for this option to have any effect.

See “**General Syntax for Attribute Options**” on page 1519 for the syntax on using a **STYLE-ELEMENT**.

“**Line Options**” on page 1522 for available **LINE-OPTIONS**.
**COLUMNDATARANGE=DATA | UNION | UNIONALL**

specifies how the X-axis data ranges of graphs within the lattice columns are scaled.

**DATA**
scales the X-axis data ranges separately for each cell in the lattice.

**UNION**
scales the X-axis data ranges separately for each column in the lattice.

This setting is supported only if all plots across the column can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**UNIONALL**
scales the X-axis data ranges across all columns in the lattice.

This setting is supported only if all plots across the column can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**Default** DATA

**Interactions**
The data ranges are scaled only if the data values are continuous and the graphs have the same orientation. If graphs cannot use the same X-axis range or if COLUMNDATARANGE=DATA, then the X-axis range of each graph is determined from the data.

If any plot statement in any cell contains a XAXIS=X2 option, then that plot's X values are ignored whenever COLUMNDATARANGE= is set to UNION or UNIONALL.

**Tips**
If column axes are used and if a lattice cell contains a LAYOUT OVERLAY statement with the XAXISOPTS= option specified, then the XAXISOPTS= option is ignored. In that case, use the COLUMNAXIS statement to specify the desired column X-axis features. For more information, see “Row and Column Axis Statements” on page 136.

By default, each cell has its own axes, which are managed using axis options in the layout statement for that cell. To consolidate the cell X axes into column X axes, set this option to UNION or UNIONALL. Then in the LAYOUT LATTICE statement block, specify a COLUMNAXES block that includes a COLUMNAXIS statement for each column X axis that needs to be managed.

**COLUMN2DATARANGE=DATA | UNION | UNIONALL**

specifies how the X2-axis data ranges of graphs within the lattice columns are scaled.

**DATA**
scales the X2-axis data ranges separately for each cell in the lattice.

**UNION**
scales the X2-axis data ranges separately for each column in the lattice.

This setting is supported only if all plots across the column can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**UNIONALL**
scales the X2-axis data ranges across all columns in the lattice.
This setting is supported only if all plots across the column can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

Default | DATA

Interactions

The data ranges are scaled only if the data values are continuous and the graphs have the same orientation. If graphs cannot use the same X2-axis range or if COLUMN2DATARANGE=DATA, then the X2-axis range of each graph is determined from the data.

If any plot statement in any cell contains a XAXIS=X option, then that plot’s X values are ignored whenever COLUMN2DATARANGE= is set to UNION or UNIONALL.

Tips

By default, each cell has its own axes, which are managed using axis options in the layout statement for that cell. To consolidate the cell X2 axes into column X2 axes, set this option to UNION or UNIONALL. Then in the LAYOUT LATTICE statement block, specify a COLUMN2AXES block that includes a COLUMNAXIS statement for each column X2 axes that needs to be managed.

If column X2 axes are used and if a lattice cell contains a LAYOUT OVERLAY statement with the X2AXISOPTS= option specified, then the X2AXISOPTS= option is ignored. In that case, use one or more COLUMNAXIS statements to specify the desired column X2-axis features. For more information, see “Row and Column Axis Statements” on page 136.

COLUMNGUTTER=\textit{dimension}  
specifies the amount of empty space that is between the columns.

Default | 0

Note  
If there are \( n \) columns, then there are \( n-1 \) gutters.

See  
“\textit{dimension}” on page 1480

COLUMNS=\textit{integer}  
specifies the number of columns in the lattice.

Defaults

If ORDER= ROWMAJOR, then the default is 1.

If ORDER=COLUMNMAJOR, then as many columns are created as are needed to satisfy the ROWS= request.

Interactions

If both ROWS=\( n \) and COLUMNS=\( m \) are specified, then an \( n \) by \( m \) grid of cells is created. If the number of statements that define cell contents is greater than \( n \times m \), then the grid size does not expand and some statements are not displayed. If the number of statements that define cell contents is less than \( n \times m \), then the grid will contain empty cells.

If this option is not defined and ORDER=COLUMNMAJOR, then the number of columns is dynamically determined by the number of defined cells.
**COLUMNWEIGHTS=** | **PREFERRED** | (**weight-list**)
--- | --- | ---
specifies the proportional width of each column relative to the overall grid width, not including the headings and sidebars.

**UNIFORM**
equally divides the total available width among all of the columns.

**PREFERRED**
specifies that each column gets its preferred width.

Preferred widths are determined as follows:

- Columns that contain one or more vertically one-dimensional plots get the maximum preferred width from the vertically one-dimensional plots.
- The remaining columns that do not contain vertically one-dimensional plots get an equal amount of width from the remaining space.

**Interaction**
If a one-dimensional box plot is specified in the preferred column, then the box plot's BOXWIDTH= option is ignored.

**Notes**
The PREFERRED option is used for LATTICE layouts that mix one-dimensional and two-dimensional plots in the grid. It enables the layout to compute the weights automatically for columns that contain one-dimensional plots.

Examples of one-dimensional plots include axis tables, block plots, fringe plots, and box plots with Y= values only.

**(** **weight-list**)
a space-separated list of column weights.

The list should contain a weight for each column, which can be expressed as one of the following:

**PREFERRED**
specifies that the corresponding column gets its preferred width as described previously.

**Note**
The PREFERRED option should be used on columns that contain only one-dimensional plots. Using the PREFERRED keyword on columns that contain a mix of one-dimensional and two-dimensional plots might cause unexpected results. In that case, use a numeric weight instead.

**Tip**
The PREFERRED option is particularly useful for axis tables.

**number**
specifies that the corresponding column gets a width that is based on the proportion of the specified number to the total of the numbers in the weight list.

For example, the following weight specifications are equivalent:

```
columnweights=(0.2 0.3 0.5)
columnweights=(2 3 5)
```

In these examples, the first column gets 20% of the available width, the second column gets 30%, and the third column gets 50%.
If the list contains the PREFERRED keyword, then the number specifies the proportion of the width that remains after the preferred width or widths are calculated and subtracted from the available width.

**Requirements**  
The values in the weight list must be enclosed in parentheses.

- If there are \( n \) columns in the grid, then the weight list should contain \( n \) weights, one for each column.
- All of the numbers in the list must be positive.
- At least one number in the list must be nonzero.

**Note**  
The weights for all of the columns are specified as a proportion and, as such, are not required to total 1.0.

**Example**  
Here is an example that specifies the preferred width for the first column, 20\% of the remaining width to the second column, and 80\% of the remaining width to the third column.

\[
\text{columnweights=(preferred 0.2 0.8)}
\]

**HALIGN=** CENTER | LEFT | RIGHT | number  
specifies this layout’s horizontal alignment within its parent when nested within an overlay-type or REGION layout.

- **number** specifies the horizontal alignment as a fraction of the parent container’s width.

  **Range**  
  0 to 1, where 0 is all the way to the left and 1 is all the way to the right.

  **Interaction**  
  For a **number** setting to take effect, LOCATION=INSIDE must be set. A **number** setting is invalid on this option when LOCATION=OUTSIDE.

  **Default**  
  CENTER

  **Restriction**  
  This option has effect only when this layout is nested within a REGION layout, or when this layout is nested in an overlay-type layout and AUTOALIGN=NONE.

**See**  
“LAYOUT OVERLAY Statement” on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

**Example**  
In the following example, the LATTICE layout is the child of the OVERLAY layout and is positioned in the OVERLAY’s top right corner.

\[
dynamic \text{VAR STAT1 STAT2 STAT3;} 
\text{layout overlay;} 
\text{histogram VAR;} 
\text{layout lattice / VALIGN=TOP HALIGN=RIGHT} 
\text{columns=1;} 
\text{entry STAT1;} 
\text{entry STAT2;} 
\text{entry STAT3;} 
\text{endlayout;} 
\]
endlayout;

**OAQUE=TRUE | FALSE**
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

**Default** FALSE

**Interaction** When this option is set to FALSE, the background color is not used.

**See** “boolean” on page 1479 for other Boolean values that you can use.

**ORDER=ROWMAJOR | COLUMNMAJOR**
specifies whether data cells are populated by column priority or by row priority.

**ROWMAJOR**
fills all the columns in a row, from left to right, before going to the next row.

**COLUMNMAJOR**
fills all the rows in a column, from top to bottom, before going to the next column.

**Default** ROWMAJOR

**Requirements**
- If this option is set to COLUMNMAJOR, then the ROWS= option must be specified to indicate how many rows to fill before wrapping to the next column. The default number of rows is 1.
- If this option is set to ROWMAJOR, then the COLUMNS= option must be specified to indicate how many columns to fill before wrapping to the next column. The default number of columns is 1.

**OUTERPAD=AUTO | dimension | (pad-options)**
specifies the amount of extra space to add outside the layout border.

**AUTO**
specifies that the default outside padding for this component is used.

**dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

**(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can contain one or more of the following pad options:

**LEFT=dimension**
specifies the amount of extra space to add to the left side.

**RIGHT=dimension**
specifies the amount of extra space to add to the right side.

**TOP=dimension**
specifies the amount of extra space to add to the top.

**BOTTOM=dimension**
specifies the amount of extra space to add to the bottom.

**Note** Sides that are not assigned padding are padded with the default amount.
Tip Use *pad-options* to create non-uniform padding.

Default **AUTO**

Note The default units for *dimension* are pixels.

See “*dimension*” on page 1480

**PAD=*dimension* | (pad-options)**

specifies the amount of extra space that is added inside the layout border.

*dimension*

specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

(pad-options)

a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

**LEFT=*dimension***

specifies the amount of extra space added to the left side.

Default 0

**RIGHT=*dimension***

specifies the amount of extra space added to the right side.

Default 0

**TOP=*dimension***

specifies the amount of extra space added to the top.

Default 0

**BOTTOM=*dimension***

specifies the amount of extra space added to the bottom.

Default 0

Note Sides that are not assigned padding are padded with the default amount.

Tip Use *pad-options* to create non-uniform padding.

Note The default units for *dimension* are pixels.

See “*dimension*” on page 1480

**ROWDATARANGE=DATA | UNION | UNIONALL**

specifies how the Y-axis data ranges of graphs within the lattice rows are scaled.

**DATA**

scales the Y-axis data ranges separately for each cell in the lattice.

**UNION**

scales the Y-axis data ranges separately for each row in the lattice.
This setting is supported only if all plots down the row can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**UNIONALL**

scales the Y-axis data ranges across all rows in the lattice.

This setting is supported only if all plots down the row can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**Default**

DATA

**Interactions**

The data ranges are scaled only if the data values are continuous and the graphs have the same orientation. If graphs cannot use the same vertical range or if ROWDATARANGE=DATA, then the vertical range of each graph is determined from the data.

If any plot statement in any cell contains a YAXIS=Y2 option, then that plot's Y values are ignored whenever ROWDATARANGE= is set to UNION or UNIONALL.

**Tips**

By default, each cell has its own axes, which are managed using axis options in the layout statement for that cell. To consolidate the cell Y axes into a row Y axis, set this option to UNION or UNIONALL. Then in the LAYOUT LATTICE statement block, specify a ROWAXES block that contains a ROWAXIS statement for each row Y axis that needs to be managed.

If row axes are used and if a lattice cell contains a LAYOUT OVERLAY with the YAXISOPTS= option specified, then the YAXISOPTS option is ignored. In that case, use one or more ROWAXIS statements to specify desired row Y-axis features. For more information, see “Row and Column Axis Statements” on page 136.

**ROW2DATARANGE=DATA | UNION | UNIONALL**

specifies how the Y2-axis data ranges of graphs within the lattice rows are scaled.

**DATA**

scales the Y2-axis data ranges separately for each cell in the lattice.

**UNION**

scales the Y2-axis data ranges separately for each row in the lattice.

This setting is supported only if all plots down the row can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**UNIONALL**

scales the Y2-axis data ranges across all rows in the lattice.

This setting is supported only if all plots down the row can share the same data range and axis type. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

**Default**

DATA

**Interactions**

The Y2-axis data ranges are scaled only if the data values are continuous and the graphs have the same orientation. If graphs
cannot use the same vertical range or if ROW2DATARANGE=DATA, then the vertical range of each graph is determined from the data.

If any plot statement in any cell contains a YAXIS=Y2 option, then that plot's Y values are ignored whenever ROW2DATARANGE= is set to UNION or UNIONALL.

**Tips**

By default, each cell has its own axes, which are managed using axis options in the layout statement for that cell. To consolidate the cell Y axes into row X2 axes, set this option to UNION or UNIONALL, and then in the LAYOUT LATTICE statement block, specify a ROW2AXES block that contains a ROWAXIS statement for each row Y2 axis that needs to be managed.

If row Y2 axes are used and if a lattice cell contains a LAYOUT OVERLAY with the Y2AXISOPTS= option specified, then the Y2AXISOPTS option is ignored. In that case, use one or more ROWAXIS statements to specify the desired row Y2-axis features. For more information, see "Row and Column Axis Statements" on page 136.

**ROWGUTTER=** *dimension*

specifies the amount of empty space between the rows.

**Default** 0

**Note** If there are *n* rows, then there are *n*-1 gutters.

**See** "dimension" on page 1480

**ROWS=** *integer*

specifies the number of rows in the lattice.

**Defaults** If ORDER=COLUMNMAJOR, then the default is 1.

If ORDER=ROWMAJOR, then as many ROWS are created as needed to satisfy the COLUMNS= request.

**Interactions** If both ROWS=*n* and COLUMNS=*m* are specified, then an *n* by *m* grid of cells is created. If the number of statements that define cell contents is greater than *n* x *m*, then the grid size does not expand and some statements are not displayed. If the number of statements that define cell contents is less than *n* x *m*, then the grid will contain empty cells.

If ORDER=ROWMAJOR and ROWS= is not defined, then the number of rows is dynamically determined by the number of defined cells.

**ROWWEIGHTS=** **UNIFORM | PREFERRED | (weight-list)**

specifies the proportional height of each row relative to the overall grid height, not including the headings and sidebars.

**UNIFORM**

equally divides the total available height among all of the rows.

**PREFERRED**

specifies that each row gets its preferred height.
Preferred height is determined as follows:

- Rows that contain one or more horizontally one-dimensional plots get the maximum preferred height from the horizontally one-dimensional plots.
- The remaining rows that do not contain horizontally one-dimensional plots get an equal amount of height from the remaining space.

**Notes**  
The PREFERRED option is used for LATTICE layouts that mix one-dimensional and two-dimensional plots in the grid. It enables the layout to compute the weights automatically for rows that contain one-dimensional plots.

Examples of one-dimensional plots include axis tables, block plots, fringe plots, and box plots with Y= values only.

*(weight-list)*  
a space-separated list of row weights.

The list should contain a weight for each row, which can be expressed as one of the following:

**PREFERRED**  
specifies that the corresponding row gets its preferred height as described previously.

*Note*  
The PREFERRED option should be used on rows that contain only one-dimensional plots. Using the PREFERRED keyword on rows that contain a mix of one-dimensional and two-dimensional plots might cause unexpected results. In that case, use a numeric weight instead.

**number**  
specifies that the corresponding row gets a height that is based on the proportion of the specified number to the total of the numbers in the weight list.

For example, the following weight specifications are equivalent:

```
rowweights=(0.2 0.3 0.5)
rowweights=(2 3 5)
```

In these examples, the first row gets 20% of the available height, the second row gets 30%, and the third row gets 50%.

If the list contains the PREFERRED keyword, then the number specifies the proportion of the height that remains after the preferred height or heights are calculated and subtracted from the available height.

**Requirements**  
The values in the weight list must be enclosed in parentheses.

If there are $n$ rows in the grid, then the weight list should contain $n$ weights, one for each row.

All of the numbers in the list must be positive.

At least one number in the list must be nonzero.

*Note*  
The weights for all of the rows are specified as a proportion and, as such, are not required to total 1.0.
Example

Here is an example that specifies 25% of the available height for the first row, 25% for the second row, and 50% for the third row.

```
rowweights=(1 1 2)
```

**SHRINKFONTS**=TRUE | FALSE

specifies whether fonts in the layout are scaled down.

**Default**

FALSE

**Note**

If SHRINKFONTS=TRUE, the font scaling applies to all child layouts within the parent layout. If a child layout also specifies SHRINKFONTS=TRUE, the fonts for the children of that child layout are further scaled down.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**SKIPEMPTYCELLS**=TRUE | FALSE

specifies whether row and column axes skip the unused cells in a partially filled lattice.

FALSE

displays row and column axes at their normal locations.

TRUE

skips empty cells and “snaps” the row and column axes to the nearest populated cell, both vertically and horizontally.

**Default**

FALSE

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**VALIGN**=CENTER | TOP | BOTTOM | number

specifies this layout’s vertical alignment within its parent when nested within an overlay-type or REGION layout.

**number**

specifies the vertical alignment as a fraction of the parent container’s height.

**Range**

0 to 1, where 0 is on the bottom and 1 is on the top.

**Interaction**

For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

**Default**

CENTER

**Restriction**

This option has effect only when this layout is nested within a REGION layout, or when this layout is nested in an overlay-type layout and AUTOALIGN=NONE.

**See**

“LAYOUT OVERLAY Statement” on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

**Example**

In the following example, the LATTICE layout is the child of the OVERLAY layout. The child layout appears in either the top right or top left position, based on which position has more “unoccupied” area.
dynamic VAR STAT1 STAT2 STAT3;
layout overlay / height=500px width=600px;
    histogram VAR;
layout lattice / VALIGN=TOP HALIGN=RIGHT
    height=80px width=70px columns=1;
    entry STAT1;
    entry STAT2;
    entry STAT3;
endlayout;
endlayout;

SIDEBAR Optional Arguments

**ALIGN=TOP | BOTTOM | LEFT | RIGHT**
specifies the alignment of the sidebar.

- **Default** BOTTOM
- **Tip** You can use **ENTRY** statements to place rotated text in the right or left side bar.

**SPACEFILL=TRUE | FALSE**
specifies whether to fill all the area of the sidebar with its contents.

- **Default** TRUE
- **Tip** To prevent a layout block within the sidebar from expanding to the sidebar boundaries, set this option to FALSE.
- **See** “boolean ” on page 1479 for other Boolean values that you can use.

Details

Statement Description

The LAYOUT LATTICE statement creates a grid of graphs that are aligned across rows and columns. For plot statements that are specified in the layout block or nested in a LAYOUT OVERLAY statement, the LATTICE layout automatically aligns the plot areas and tick display areas in the plots.

**Note:** To achieve the alignment, the LATTICE layout automatically aligns plot areas and tick display areas across rows and columns. Also, it overrides axis offset settings in the OVERLAY layouts that you specify in those rows and columns. (For details about offsets and the tick display area, see “Adjusting Axis Offsets” on page 985.) If you do not want this alignment, then you might use LAYOUT GRIDDED instead. For example, if you have a heterogeneous panel of graphs, such as a mix of scatter plots, box plots, bar charts, or other types of graphs, then you might consider using LAYOUT GRIDDED rather than LAYOUT LATTICE.

The layout can unify the scale of the data ranges that are displayed in the plots, based on the values set for the COLUMNDATARANGE= and ROWDATARANGE= options. If one or more plots within the template use the XAXIS= option to produce independent X2 (top) axes, then the X2 data scales can be unified, based on the values set for the COLUMN2DATARANGE= option. If one or more plots within the template use the YAXIS= option to produce independent Y2 (right) axes, then the
Y2 data scales can be unified, based on the values set for the `ROW2DATARANGE=` options. The data ranges can be scaled separately for each column, for each row, or for both. Or they can be scaled across all columns, all rows, or all of both.

When the data-range scales are unified, you can simplify the lattice by displaying only the row and column axes that apply to all of the graphs across the corresponding columns or rows. See “Row and Column Axis Statements” on page 136 for more details.

The following figure shows the parts of the LATTICE layout with the default axis display (internal axes are displayed).

This next figure shows the parts of the LATTICE layout when the graph display is simplified so that only row and column axes are displayed.

Note: The figure shows secondary X (top) and secondary Y (right) axes. The layout also enables you to generate independent X2 (top) and independent Y2 (right) axes. For details, see “Row and Column Axis Statements” on page 136.
The rows and columns can be separated by areas called “gutters,” which are controlled by the `COLUMNGUTTER=` and `ROWGUTTER=` options. In addition, the `COLUMNWEIGHTS=` and `ROWWEIGHTS=` options can be used to allocate a proportion of available space to each row and column.

The LATTICE layout automatically decides how much area to allocate to cell contents:

- Text items have a fixed size based on the amount of text and the font properties.
- Graphs take up the remaining space.

The layout’s grid size is determined by the `COLUMNS=` and `ROWS=` options.

By default, the results of the `GTL-statements` are placed into the grid sequentially from left to right, wrapping to a new row each time the current row is filled. You can use the `ORDER=` option to fill cells from top to bottom down a column. In that case, the lattice cells wrap to a new column each time the current column is filled.

Cell Contents

The following general syntax is used to define the contents of each cell in a `LAYOUT LATTICE`:

```
GTL-statement(s) | cell-statement-block(s)
```

A `cell-statement-block`, when used, has the following syntax:

```
CELL;
```
The following guidelines apply to defining cell content:

- The contents of each cell is generated by GTL statements, which can be specified independently or enclosed in a CELL block.

- Independent GTL statements include text statements, plot statements, or layout blocks. Each independently specified GTL statement or layout block provides content for one cell.

- A CELL block can include text statements, plot statements, or layout blocks. Each CELL block provides content for one cell.

- Within a CELL block, you can use a CELLHEADER block to generate one or more header lines within the cell. Specify each header line on a separate GTL statement within the CELLHEADER block. The header block is typically used to specify one or more text statements, but other statements are allowed within the block. For example, you can specify a LAYOUT GRIDDED statement to produce a grid of text for the header.

- You can use only one CELLHEADER block per CELL block. If you specify more than one, then only the last one is used.

- If you do not specify a CELLHEADER block in a CELL block, then the enclosed GTL statements produce the same results that they would produce if they were specified independently.

- If more than one plot statement is needed to generate contents for a cell, you should place the plot statements in a layout block such as LAYOUT OVERLAY. Otherwise, unexpected results might occur. This applies to independent GTL statements and to GTL statements in a CELL block. See Figure 4.3 on page 136.

If a CELL block, or an independent statement or layout block for a lattice cell does not produce any output, then the space for that cell might not be retained as an empty cell in the lattice. In that case, the empty cell is removed, and the remaining cells (if any) fill the gap in the lattice. A statement produces no output when required data for that statement does not resolve. A layout block produces no output when it contains no statements or when none of the statements contained in the block produce any output.

The example code shows a LAYOUT LATTICE block that uses one GTL statement and one CELL block to generate the two-column lattice shown in the following figure:
Row and Column Axis Statements

Overview

By default, each cell in the lattice has its own internal axes, which are managed using axis options in the layout statement for that cell. Here is an example of a lattice with internal axes.

```sas
proc template;
  define statgraph cellcontents;
    begingraph;
      layout lattice /
        columngutter=5 columns=2;

        /* independent plot statement - defines first cell */
        scatterplot x=age y=height;

        /* cell block - defines second cell */
        cell;
          cellheader;
            entry "Cell Header" / border=true;
          endcellheader;
          /* two plot statements are needed - enclose
            the statements in a LAYOUT OVERLAY block */
          layout overlay;
            scatterplot x=weight y=height;
            referenceline y=53 / lineattrs=(pattern=shortdash)
              curvelabellocation=inside curvelabel="Reference Line";
          endlayout;
        endcell;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.class template=cellcontents;
run;
```
Notice that the axes ranges for each plot in the lattice differ. If the data ranges for each plot are comparable, you can unify the axes to a common scale for all of the plots in the lattice. In that case, you can also simplify and clarify your lattice by creating row and column axes that are shared by the cells in the lattice. Here is an example of a lattice with row and column axes.

Row and column axes are defined by row and column axis statement blocks in the LAYOUT LATTICE statement block. Row and column axis statement blocks are useful only if the data ranges across the affected columns or rows are comparable and can be unified to a common scale. For example, row and column axes are not supported if an affected lattice cell contains a LAYOUT OVERLAYEQUATED statement. If the axis ranges are not unified for the affected columns or rows, then the row and column axis statement blocks in the LAYOUT LATTICE statement block are ignored.

Use the options shown in the following table to unify the axis data ranges in the lattice.
<table>
<thead>
<tr>
<th>Axis</th>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>COLUMNNDATARANGE=</td>
</tr>
<tr>
<td>X2</td>
<td>COLUMN2DATARANGE=</td>
</tr>
<tr>
<td>Y</td>
<td>ROWDATARANGE=</td>
</tr>
<tr>
<td>Y2</td>
<td>ROW2DATARANGE=</td>
</tr>
</tbody>
</table>

Specifying Row and Column Axis Features

For columns, the features for the column X axes (bottom) are specified within a COLUMNAXES block, which contains one COLUMNAXIS statement for each column X axis that needs to be manage. The COLUMNAXIS statement provides a DISPLAYSECONDARY= option, which enables you to display a column secondary X (top) axis that mirrors the column primary X axis but can have different display features. In that case, the axis features that you specify in the COLUMNAXIS statement apply to both the column primary and column secondary X axes.

If one or more plots within the LAYOUT LATTICE statement block use the XAXIS=X2 option to produce independent X2 (top) axes, then axis features for the column X2 axes are specified within a COLUMN2AXES block, nesting one COLUMNAXIS statement for each column that contains an X2 axis that needs to be managed. Within the COLUMN2AXES block, the COLUMNAXIS statement’s DISPLAYSECONDARY= option enables you to display a column secondary X2 (bottom) axis that mirrors the column primary X2 axis but can have different display features. Here again, the axis features that you specify in the COLUMNAXIS statement apply to both the column primary and the column secondary X2 axes.

Note: If you specify independent X and X2 scales at the same time, then the DISPLAYSECONDARY= option is ignored in the COLUMNAXIS statement. This is true whether the COLUMNAXIS statement is specified in a COLUMNAXES or COLUMN2AXES block.

For both the COLUMNAXES and COLUMN2AXES blocks, if a lattice cell within the column contains a LAYOUT OVERLAY statement with the XAXISOPTS= or X2AXISOPTS= option specified, these LAYOUT OVERLAY statement options are ignored. In that case, specify the desired axis features in one or more COLUMNAXIS statements in the COLUMNAXES block.

For rows, axis features for the row Y axes (left) are specified within a ROWAXES block, nesting one ROWAXIS statement for each row that contains a Y axis that needs to be managed. The ROWAXIS statement provides a DISPLAYSECONDARY= option, which enables you to display a row secondary Y (right) axis that mirrors the row primary Y axis but can have different display features. In that case, the axis features that you specify in the ROWAXIS statement apply to both the row primary and row secondary Y axes.

If one or more plots within the LAYOUT LATTICE statement block use the YAXIS= option to produce independent Y2 (right) axes, then axis features for the row Y2 axes (right) are specified within a ROW2AXES block, nesting one ROWAXIS statement for each row that contains a Y2 axis that needs to be managed. Within the ROW2AXES block, the ROWAXIS statement’s DISPLAYSECONDARY= option enables you to display a row secondary Y2 (left) axis that mirrors the row primary
Y2 axis but can have different display features. Here again, the axis features that you specify in the ROWAXIS statement apply to both the row primary and row secondary Y2 axes.

**Note:** If you specify independent Y and Y2 scales at the same time, then the DISPLAYSECONDARY= option is ignored in the ROWAXIS statement. This is true whether the ROWAXIS statement is specified in a ROWAXES or ROW2AXES block.

For both the ROWAXES and ROW2AXES blocks, if a lattice cell within the row contains a LAYOUT OVERLAY statement with the YAXISOPTS= or Y2AXISOPTS= option specified, these LAYOUT OVERLAY statement options are ignored. In that case, specify the desired axis features in one or more ROWAXIS statements in the ROWAXES block.

**Syntax and Restrictions for the Axis Statements**

The following table shows the general syntax for row and column axis-statement blocks.

<table>
<thead>
<tr>
<th>Row or Column X and Y Axes</th>
<th>Row or Column X2 and Y2 Axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMNAXES;</td>
<td>COLUMN2AXES;</td>
</tr>
<tr>
<td>COLUMNAXIS / axis-options;</td>
<td>COLUMNAXIS / axis-options;</td>
</tr>
<tr>
<td>&lt;...COLUMNAXIS-n;&gt;</td>
<td>&lt;...COLUMNAXIS-n;&gt;</td>
</tr>
<tr>
<td>ENDCOLUMNAXES;</td>
<td>ENDCOLUMN2AXES;</td>
</tr>
<tr>
<td>ROWAXES;</td>
<td>ROW2AXES;</td>
</tr>
<tr>
<td>ROWAXIS / axis-options;</td>
<td>ROWAXIS / axis-options;</td>
</tr>
<tr>
<td>&lt;...ROWAXIS-n;&gt;</td>
<td>&lt;...ROWAXIS-n;&gt;</td>
</tr>
<tr>
<td>ENDRWAXES;</td>
<td>ENDRW2AXES;</td>
</tr>
</tbody>
</table>

In the LAYOUT LATTICE block, the following restrictions apply:

- If the LAYOUT LATTICE statement sets the row or column axis data range to DATA, then the corresponding axes block is ignored. The data range must be set to UNION or UNIONALL to create row and column axes.

- Only one COLUMNAXES block can be used to manage column X axes, and only one COLUMN2AXES block can be used to manage column X2 axes. If more than one of either block is specified, then only the last one of that block type is used.

- Within a COLUMNAXES or COLUMN2AXES block, one COLUMNAXIS statement should be specified for each column that contains axes that you need to manage. Both axes blocks can contain a COLUMNAXIS statement for the same column. For example, to manage the axes in the first column of the lattice, the COLUMNAXES block can contain a COLUMNAXIS statement that manages the column’s X axes. The COLUMN2AXES block can contain a COLUMNAXIS statement that manages the column’s X2 axes.

- Only one ROWAXES block can be used to manage row Y axes, and only one ROWAXES block can be used to manage row Y2 axes. If more than one of either block is specified, then only the last one of that block type is used.
Within a ROWAXES or ROW2AXES block, one ROWAXIS statement should be specified for each row that contains axes that need to be managed. Both axes blocks can contain a ROWAXIS statement for the same row. For example, to manage the axes in the first row of the lattice, the ROWAXES block can contain a ROWAXIS statement that manages the row’s Y axes. The ROW2AXES block can contain a ROWAXIS statement that manages the column’s Y2 axes.

If the number COLUMNLAXIS or ROWAXIS statements is greater than the number needed, then the extra statements are ignored. If the number of statements is fewer than the number needed, then the additional COLUMNAXIS or ROWAXIS statements are automatically generated with DISPLAY=NONE option in effect for each.

For the list of axis-options, see “Row and Column Axis Options for LAYOUT LATTICE” on page 1070.

The following example shows a LAYOUT LATTICE block that uses a ROWAXES block to set the row Y axis and to display grid lines for the row display.

```
begingraph;
  layout lattice /
    rowdatarange=union
    columns=2;

  /* axis definitions */
  rowaxes;
    rowaxis /griddisplay=on;
  endrowaxes;

  /* cell contents */
  scatterplot x=x y=t;
  scatterplot x=x y=y;

  endlayout;
endgraph;
```

Here, the LAYOUT LATTICE statement specifies the ROWDATARANGE option to unify the data ranges across rows in the lattice. Because LAYOUT LATTICE specifies COLUMNS=2 and there are two plot statements in the template, the resulting graph has two columns and only one row. Thus, only one ROWAXIS statement is needed in the ROWAXES block to specify axis attributes for that row of graphs. A ROW2AXES block is not needed because neither SCATTERPLOT statement in the template maps data to the Y2 axis.

For more information and examples that demonstrate how data are mapped to the axes, see “Plot Data Are Mapped to a Designated Axis” on page 974.

Header Statements

Header statements are used to display one or more headers for the rows and columns in a LATTICE layout. Each statement is specified as a block in the form statement - ENDSstatement. The header block is typically used to specify one or more text statements, but other statements are allowed within the block. For example, you could specify a LAYOUT GRIDDED statement to produce a grid of text for the header.

The general syntax for a COLUMNHEADERS statement is

```
COLUMNHEADERS;
  GTL-statement(s);
```
The following header statements are available:

<table>
<thead>
<tr>
<th>Header Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMNHEADERS</td>
<td>specifies a header for the primary (bottom) column-header position.</td>
</tr>
<tr>
<td>COLUMN2HEADERS</td>
<td>specifies a header for the secondary (top) column-header position.</td>
</tr>
<tr>
<td>ROWHEADERS</td>
<td>specifies a header for the primary (left) row-header position. ENTRY statements can be used to specify rotated text.</td>
</tr>
<tr>
<td>ROW2HEADERS</td>
<td>specifies a header for the secondary (right) row-header position. ENTRY statements can be used to specify rotated text.</td>
</tr>
</tbody>
</table>

- The LAYOUT LATTICE aligns headers with the columns, or the rows, or both.
- Each of the header blocks COLUMNHEADERS, COLUMN2HEADERS, ROWHEADERS, and ROW2HEADERS can be used once in a LAYOUT LATTICE block. If more than one block is specified for one of the statements, then only the last specified block for that statement is used.

The following example shows a LAYOUT LATTICE block that uses a COLUMNHEADERS block to display column headers above the left and right columns in the lattice.

```
begingraph;
  layout lattice / columns=2;

  /* Lattice header definitions */
  columnheaders;
    entry "Left Column";
    entry "Right Column";
  endcolumnheaders;

  /* cell contents */
  scatter x=x y=t;
  scatter x=x y=y;

  endlayout;
endgraph;
```

**Sidebar Statements**

A LAYOUT LATTICE supports the display of a sidebar between a row or column header and a row or column axis. (See the figures in “Statement Description” on page 132.) A sidebar spans across columns or rows and is useful for displaying information that applies to all of the columns or all of the rows. For example, sidebars are useful for displaying legends.

A SIDEBAR statement has the following syntax:
You can specify up to four SIDEBAR blocks in a LAYOUT LATTICE, one for each of the top, bottom, left, and right sidebar positions.

If two or more SIDEBAR blocks have the same alignment, then the sidebar information forms two or more columns (ALIGN=LEFT or ALIGN=RIGHT) within the sidebar area. Or it forms two or more rows (ALIGN=TOP or ALIGN=BOTTOM) within the sidebar area.

Only one statement (such as ENTRY or DISCRETELEGEND) or one layout block (such as LAYOUT GRIDDED) is allowed in a SIDEBAR block. To create multi-line text in a sidebar, nest ENTRY statements within a LAYOUT GRIDDED block.

The LAYOUT LATTICE automatically aligns a sidebar with the lattice columns or rows.

The following example shows a LAYOUT LATTICE block that uses a SIDEBAR block to display a top sidebar in the lattice.

```
begingraph;
layout lattice / columns=2;

sidebar / align=top;
    layout gridded / border=true;
    entry "Top Sidebar";
    entry "(spans both columns)";
endlayout;
endsidebar;

scatterplot x=x y=t;
scatterplot x=x y=y;
endlayout;
begingraph;
```

Examples:

Example 1: Lattice with Internal Axes

This example shows you how to create a four-cell lattice of graphs (two rows, two columns), where each graph has its own, independently managed X and Y axes. The following features are used in this example:

- The following LAYOUT LATTICE statement options:
  - The COLUMNS= option specifies the number of columns in the lattice.
  - The COLUMNGUTTER= and ROWGUTTER= options specify the spacing between the lattice cells.
- A COLUMN2HEADER block in the LAYOUT LATTICE statement block displays headers at the top of each column. ENTRY statements provide the text for each header.
Four LAYOUT OVERLAY statement blocks in the LAYOUT LATTICE statement block provide the content for each cell.

Axis options in each LAYOUT OVERLAY statement manage the axes for each plot.

Example Graph

The following graph was generated by the “Example Program” on page 143:

Example Program

/* Create the stock data for Microsoft for the years 2001 and 2002 */
data stockyear1 stockyear2;
  set sashelp.stocks(where=(stock eq "Microsoft" and year(date) in (2001 2002)));
  volume = volume/1000000;
  if year(date) = 2001 then
    output work.stockyear1;
  else if year(date)=2002 then
    output work.stockyear2;
run;

data msstock;
  merge work.stockyear1(rename=(date=year1date close=year1close volume=year1vol))
    work.stockyear2(rename=(date=year2date close=year2close volume=year2vol));
  format year1date year2date monname3. year1close year2close dollar6.;
run;

/* Create the graph template */
proc template;
  define statgraph graphlattice;
Example 2: Lattice with Row and Column Axes

In the lattice shown in “Example 1: Lattice with Internal Axes” on page 142, because the axes in each cell are independently managed, the axis ranges can vary across the plots. However, the axis ranges in each plot in that lattice are comparable. In that case, you can use options COLUMNDATARANGE= and ROWDATARANGE= in the LAYOUT LATTICE statement to create uniform axes for the plots in individual columns or rows, or for all of the plots in all of the columns or rows. The axes are
scaled to accommodate the data ranges in all of the included plots. In addition, you can use COLUMNAXES and ROWAXES statement blocks in the LAYOUT LATTICE statement block to consolidate the individual internal axes into common row and column axes. This example shows you how to modify the lattice in “Example 1: Lattice with Internal Axes” on page 142 to use common row and column axes for the plots in the lattice. The following features are used in this example:

- the following LAYOUT LATTICE statement options:
  - The COLUMNS= option specifies the number of columns in the lattice.
  - The COLGUTTER= and ROWGUTTER= options specify the spacing between the lattice cells.
  - The COLMONGODB=UNION and ROWMONGODB=UNION options specify uniform axes for each row and column.
- A COLUMN2HEADER block in the LAYOUT LATTICE statement block displays headers at the top of each column, and ENTRY statements provide the header text.
- A COLUMNAXES statement block in the LAYOUT LATTICE statement block specifies COLUMNAXES statements, which manage each column axis. Likewise, a ROWAXES statement block provides ROWAXES statements, which manage each row axis.
- Four LAYOUT OVERLAY statement blocks in the LAYOUT LATTICE statement block provide the content for each cell. No axis options can be used in the LAYOUT OVERLAY statements in this case because axes blocks are used to manage common row and column axes.

Example Graph

The following graph was generated by the “Example Program” on page 146:
/* Create the stock data for Microsoft for the years 2001 and 2002 */
data stockyear1 stockyear2;
  set sashelp.stocks(where=(stock eq "Microsoft" and year(date) in (2001 2002)));
  volume = volume/1000000;
  if year(date) = 2001 then
    output work.stockyear1;
  else if year(date)=2002 then
    output work.stockyear2;
run;

data msstock;
  merge work.stockyear1(rename=(date=year1date close=year1close volume=year1vol))
    work.stockyear2(rename=(date=year2date close=year2close volume=year2vol));
  format year1date year2date monname3. year1close year2close dollar6.;
r
/* Create the graph template */
proc template;
  define statgraph graphlattice;
  begingraph;
    entrytitle "Microsoft Stock Performance in 2001 and 2002";
    layout lattice / columns=2 columndatarange=union columngutter=5
      rowdatarange=union rowgutter=5;
    column2headers;
      entry textattrs=GraphData1(weight=bold size=9pt) "Year 2001";
      entry textattrs=GraphData2(weight=bold size=9pt) "Year 2002";
    endcolumn2headers;
    columnaxes;
      columnaxis / display=(ticks tickvalues)
        timeopts=(viewmax='01DEC2001'd tickvalueformat=monname3.);
      columnaxis / display=(ticks tickvalues)
        timeopts=(viewmax='01DEC2002'd tickvalueformat=monname3.);
    endcolumnaxes;
    rowaxes;
      rowaxis / label="Closing Price";
      rowaxis / label="Volume (Millions)";
    endrowaxes;
  /* Row 1 content */
  layout overlay;
    seriesplot x=year1date y=year1close / display=all smoothconnect=true
      lineattrs=GraphData1 markerattrs=GraphData1;
  endlayout;
  layout overlay;
    seriesplot x=year2date y=year2close / display=all smoothconnect=true
      lineattrs=GraphData2 markerattrs=GraphData2;
  endlayout;
  /* Row 2 content */
  layout overlay;
    needleplot x=year1date y=year1vol / displaybaseline=off
Example 3: Lattice with Sidebar

This example shows a two-cell lattice (two columns, one row) with a legend placed in a sidebar at the top of the graph. A ROWAXES block is used in the example to create a row Y axis. The following features are used in this example:

- The ROWDATARANGE=UNION option in the LAYOUT LATTICE statement ensures that the data ranges of all Y= columns in the row cells are merged into a common axis range. This facilitates the visual comparison of the cells.
- A SIDEBAR block in the LAYOUT LATTICE statement block creates a sidebar at the top of the lattice. A DISCRETELEGEND statement populates the sidebar with a graph legend.
- The DISPLAYSECONDARY= option in the ROWAXIS statement displays a secondary Y axis on the right. The secondary Y axis is a duplicate of the primary Y axis, which makes it easier to read Y-axis values when viewing the bar chart in the right cell.

Example Graph

The following graph was generated by the “Example Program” on page 148:
Example Program

```sas
proc template;
  define statgraph layoutlattice;
  begingraph;
    entrytitle "Vehicle Gas Mileage";
    entryfootnote "Averages of 428 models from 38 manufacturers";
    layout lattice / columns=2 rowdatarange=union;
      layout overlay / cycleattrs=true;
        barchart x=origin y=mpg_highway / stat=mean barwidth=.8 name="H" ;
        barchart x=origin y=mpg_city / stat=mean barwidth=.5 name="C" ;
      endlayout;
      layout overlay / cycleattrs=true;
        barchart x=type y=mpg_highway / stat=mean barwidth=.8;
        barchart x=type y=mpg_city / stat=mean barwidth=.5;
      endlayout;
    sidebar / align=top;
      discretelegend "H" "C" / border=false;
    endsidebar;
    rowaxes;
      rowaxis / display=(tickvalues) displaysecondary=(tickvalues)
                  griddisplay=on;
    endrowaxes;
  endlayout;
endgraph;
end;
run;
proc sgrender data=sashelp.cars template=layoutlattice;
run;
```
LAYOUT OVERLAY Statement

Builds a composite from one or more GTL-statements. The composite could be an entire graph. Or, if this layout is nested in a GRIDDED or LATTICE layout, then the composite typically provides contents for one cell in the parent layout.

Restrictions: You can add one or more 2-D plots to the graph area that the LAYOUT OVERLAY statement creates, but all of the graphs will share the same set of axes.

3-D plots are not allowed.

Interaction: When nested within another layout type, the OVERLAY layout defines the graph display for one cell of the parent layout. A separate OVERLAY layout is specified for each cell.

Tips: If you want equal sized units on the X and Y axes, use the LAYOUT OVERLAYEQUATED statement.

By default, the first plot specified in the LAYOUT OVERLAY block defines the axis characteristics. If you want another plot to define the axis characteristics, specify PRIMARY=TRUE in the statement for that plot.

Overlaid plots are stacked in the order in which they are specified. The last plot specified is placed on top. When plots are stacked, a plot’s graphical data might obscure the graphical data of the plots beneath it. In that case, you can change the plot order or use transparency to make all of the graphical data visible.

The LAYOUT OVERLAY block is primarily used to specify 2-D plot statements, but you can also specify other statements such as DISCRETELEGEND, CONTINUOUSLEGEND, and ENTRY.

Syntax

LAYOUT OVERLAY <options>;
  GTL-statements;
  <INNERMARGIN <options(s)>;
    block-plot-statement(s); | axis-table statement(s);
  ENDINNERMARGIN;>
  <... more-innermargin-blocks ...>>
ENDLAYOUT;

Summary of Optional Arguments

Appearance options

ASPECTRATIO=AUTO | positive-number
  specifies the aspect ratio of the plot’s wall area.

BACKGROUND_COLOR=style-reference | color
  specifies the color of the layout background.

BORDER=TRUE | FALSE
  specifies whether a border is drawn around the layout.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the border line around the layout.

CYCLEATTRS=TRUE | FALSE
specifies whether the default visual attributes of markers, lines, and fills in
nested plot statements automatically change from plot to plot.

**OPAQUE=TRUE | FALSE**
 specifies whether the layout background is opaque (TRUE) or transparent
(FALSE).

**OUTERPAD=**
 specifies the amount of extra space to add outside the layout border.

**PAD=**
 specifies the amount of extra space that is added inside the layout border.

**WALLCOLOR=**
 specifies the fill color of the plot wall area.

**WALLDISPLAY=**
 specifies whether the plot’s wall and wall outline are displayed.

**Axes options**

**X2AXISOPTS=(axis-options)**
 specifies one or more X2 axis options.

**XAXISOPTS=(axis-options)**
 specifies one or more X-axis options.

**Y2AXISOPTS=(axis-options)**
 specifies one or more Y2 axis options.

**YAXISOPTS=(axis-options)**
 specifies one or more Y-axis options.

**Optional Arguments**

**ASPECTRATIO=**
 specifies the aspect ratio of the plot’s wall area. The ratio is expressed as a
positive decimal fraction representing *wall-height* divided by *wall-width*. For
example, 0.75 is a 3/4 aspect ratio and 1.0 is a square aspect ratio.

Default: AUTO. The wall area is sized to the maximum area that can fill the
available space inside the OVERLAY layout.

Interaction: When the LAYOUT OVERLAY statement is nested in a LAYOUT
LATTICE block, the ASPECTRATIO= option is ignored unless
ROWDATARANGE=DATA, COLUMNDATARANGE=DATA,
ROW2DATARANGE=DATA, and COLUMN2DATARANGE=DATA are
in effect in the LAYOUT LATTICE statement.

See “LAYOUT LATTICE Statement” on page 118

**BACKGROUNDCOLOR=**
 specifies the color of the layout background.

**style-reference**
 specifies a style reference in the form *style-element:*style-attribute.*
 Only the style attribute named COLOR or CONTRASTCOLOR is used.

**color**
 specifies a color by name.

See color on page 1480

Default: The GraphBackground:Color style reference.
**BORDER=TRUE | FALSE**

specifies whether a border is drawn around the layout.

**Default**

FALSE

**Interaction**

If this option is set to FALSE, then the BORDERATTRS= option is ignored.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the border line around the layout.

**Default**

The GraphBorderLines style element.

**Interaction**

BORDER= TRUE must be set for this option to have any effect.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**CYCLEATTRS=TRUE | FALSE**

specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.

**FALSE**

does not cycle the default visual attributes of multiple plots.

For example, if you overlay three series plots, then each series line has the same default visual properties.

**TRUE**

attempts to use the GraphData1–GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others).

Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

**Default**

FALSE

**See**

“Rotating Visual Attributes for Each Plot in an Overlay” on page 197

“boolean” on page 1479 for other Boolean values that you can use.

**Example**

In the following example, the first three series plots are assigned line properties that are based on the GraphData1, GraphData2, and GraphData3 style elements. The fourth series plot does not participate in the attribute cycling because its LINEATTRS= option assigns a line style.

```
layout overlay / cycleattrs=true;
seriesplot x=date y=var1;
seriesplot x=date y=var2;
seriesplot x=date y=var3;
seriesplot x=date y=var4 / lineattrs=GraphReference;
```
endlayout;

**OPAQUE=TRUE | FALSE**
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

<table>
<thead>
<tr>
<th>Default</th>
<th>FALSE</th>
</tr>
</thead>
</table>

**Interaction**
When this option is set to FALSE, the background color is not used.

**See**
“boolean ” on page 1479 for other Boolean values that you can use.

**OUTERPAD=AUTO | dimension | (pad-options)**
specifies the amount of extra space to add outside the layout border.

- **AUTO**
specifies that the default outside padding for this component is used.

- **dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

- **(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

  The list can contain one or more of the following pad options:

  - **LEFT=dimension**
    specifies the amount of extra space to add to the left side.

  - **RIGHT=dimension**
    specifies the amount of extra space to add to the right side.

  - **TOP=dimension**
    specifies the amount of extra space to add to the top.

  - **BOTTOM=dimension**
    specifies the amount of extra space to add to the bottom.

**Note**
Sides that are not assigned padding are padded with the default amount.

**Tip**
Use *pad-options* to create non-uniform padding.

<table>
<thead>
<tr>
<th>Default</th>
<th>AUTO</th>
</tr>
</thead>
</table>

| Note      | The default units for *dimension* are pixels. |

**See**
“*dimension*” on page 1480

**PAD=dimension | (pad-options)**
specifies the amount of extra space that is added inside the layout border.

- **dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

- **(pad-options)**
a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

  The list can include one or more of the following pad options:
LEFT=dimension
    specifies the amount of extra space added to the left side.
    Default 0

RIGHT=dimension
    specifies the amount of extra space added to the right side.
    Default 0

TOP=dimension
    specifies the amount of extra space added to the top.
    Default 0

BOTTOM=dimension
    specifies the amount of extra space added to the bottom.
    Default 0

Note   Sides that are not assigned padding are padded with the default amount.

Tip    Use pad-options to create non-uniform padding.

Note   The default units for dimension are pixels.

See    "dimension" on page 1480

WALLCOLOR=style-reference | color
    specifies the fill color of the plot wall area.

style-reference
    specifies a style reference in the form style-element:style-attribute.
    Only the style-attribute named COLOR or CONTRASTCOLOR is used.

color
    specifies a color by name.

See    color on page 1480

Default The GraphWalls:Color style reference.

Interaction This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

WALLDISPLAY=STANDARD | ALL | NONE | (display-options)
    specifies whether the plot’s wall and wall outline are displayed.

STANDARD
displays a filled wall.
    The setting of the FRAMEBORDER= attribute of the GraphWalls style
    element determines whether the wall outline is displayed.

ALL
    displays a filled, outlined wall.
NONE
displays no wall and no wall outline.

(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

OUTLINE displays the wall outline.
FILL displays a filled wall area.

Default STANDARD

Tips
Use the WALLCOLOR= option to control the fill color of the wall.

The appearance attributes of the wall fill and outline are set by the GraphWalls style element.

XAXISOPTS=(axis-options)
specifies one or more X-axis options.

Requirements
Axis options must be enclosed in parentheses and separated by spaces.

Each option must be specified as a name = value pair, and each pair must be separated by a space.

See "Axis Options for LAYOUT OVERLAY“ on page 987 for a list of axis options.

X2AXISOPTS=(axis-options)
specifies one or more X2 axis options.

Requirements
Axis options must be enclosed in parentheses and separated by spaces.

Each option must be specified as a name = value pair, and each pair must be separated by a space.

See "Axis Options for LAYOUT OVERLAY“ on page 987 for a list of axis options.

YAXISOPTS=(axis-options)
specifies one or more Y-axis options.

Requirements
Axis options must be enclosed in parentheses and separated by spaces.

Each option must be specified as a name = value pair, and each pair must be separated by a space.

See "Axis Options for LAYOUT OVERLAY“ on page 987 for a list of options.

Y2AXISOPTS=(axis-options)
specifies one or more Y2 axis options.
Requirements

Axis options must be enclosed in parentheses and separated by spaces.

Each option must be specified as a name = value pair, and each pair must be separated by a space.

See "Axis Options for LAYOUT OVERLAY" on page 987 for a list of options.

Details

The LAYOUT OVERLAY statement builds a composite using one or more GTL-statements. You can specify one or more two-dimensional plots within the layout, provided all plots can share the same type of axes. You can also specify one or more insets, such as nested layout statements (for example, LAYOUT GRIDDED), ENTRY statements, and legend statements (for example, CONTINUOUSLEGEND or DISCRETELEGEND).

The following general logic applies to rendering the composite:

Note: The details for positioning insets also apply to insets that are specified within a LAYOUT REGION block.

- All plot statements are rendered first. Plot statement results are always rendered in the plot area. The plots are stacked on top of one another in the order in which they are specified, with the last one on top. It is possible for one plot's graphical data to obscure graphical data beneath it. You can control this by selectively ordering the plot statements, or by using transparency on the individual plots, or by doing both.

- The insets are rendered next, in the order in which they are specified. As with the plot statements, it is possible for the insets to obscure the results of other statements in the layout.

- To control the horizontal and vertical positioning of some insets, you can use the inset statement's AUTOALIGN= option, or its HALIGN= and VALIGN= options. Each nested inset determines its own relative position in the parent OVERLAY. This positioning achieves the best results for text-based insets whose size can be easily fit within an open area of the graph wall. A large text-based inset might not fit well, and an inset that contains a plot might be dropped from the display without warning when the template is executed.

- Some insets, like legends, can be positioned inside or outside of the plot area using the inset statement's LOCATION= option. The inset's AUTOALIGN= or HALIGN= and VALIGN= settings are then relative to that location.

Generally, the first specified plot determines the layout's default axis characteristics. To enable another plot to define the axis characteristics, set PRIMARY=TRUE for that plot. For more information about the default axis characteristics, see "When Plots Share Data and a Common Axis" on page 979. See also the LAYOUT OVERLAYEQUATED and the LAYOUT OVERLAY3D statements.

An OVERLAY layout can also contain an inner margin, which is a nested region at the top or bottom of the OVERLAY container. One or more inner margin plots can be specified, and each is specified within an INNERMARGIN block. Within the INNERMARGIN block, only one-dimensional plot statements such as BLOCKPLOT and AXISTABLE can be specified. See "INNERMARGIN Statement" on page 179.
Example: Simple Overlay

This example shows how to create a simple overlay of two series plots using the OVERLAY layout. The following figure shows the output.

Here is the SAS code.

```sas
data workers;
  format Date monyy5.;
  input Date monyy5. Electric Masonry;
datalines;
JAN80 320.3 293.8
FEB80 315.7 285.8
MAR80 312.6 292
APR80 306.5 299.3
MAY80 308.6 301.7
JUN80 316.3 307.9
JUL80 319.5 310.7
AUG80 326.4 314.9
SEP80 330.8 312.7
OCT80 329.3 318.5
NOV80 330.6 307.7
DEC80 327.2 296.2
JAN81 316.2 259.2
FEB81 310.1 258.8
MAR81 308.5 271.5
APR81 311.1 281
MAY81 313.6 283.7
JUN81 318.3 289.3
JUL81 321.3 291.1
AUG81 327.4 295.9
SEP81 326.7 292.7
OCT81 326.4 282.6
NOV81 322.5 275.5
```
LAYOUT OVERLAYEREQUATED Statement

Builds a composite from one or more GTL-statements. The composite could be an entire graph. Or, if this layout is nested in another layout, such as a GRIDDED layout, then the composite typically provides contents for one cell in the parent layout. In an OVERLAYEREQUATED layout, the display unit of the X axis always equals the display unit of the Y axis.

Restrictions: All overlaid plots share common X and Y axes.

3-D plots are not allowed.

The axis type for both X and Y axes is always linear and is derived directly from the data. Plot types that have discrete or binned axes cannot be used with this layout. This includes BOXPLOT, BOXPLOTPARM, BARCHARTPARM, HISTOGRAM and HISTOGRAMPARM. It also includes HEATMAP and HEATMAPPARM when at least one of the plot variables is of type character, or TRUE is in effect for at least one of the options DISCRETEX=, DISCRETEY=, XBINAXIS=, and YBINAXIS=.

You can add one or more of the following X-Y plots to the graph area that the LAYOUT OVERLAYEREQUATED statement creates: BANDPLOT, CONTOURPLOTPARM, ELLIPSE, ELLIPSEPARM, LOESSPLOTPLOT, NEEDLEPLOTPLOT, PBSPLINEPLOTPLOT, REGRESSIONPLOTPLOT, SCATTERPLOTPLOT, SERIESPLOTPLOT, STEPPLOTPLOT, or VECTORPLOTPLOT. You can also add HEATMAP and HEATMAPPARM plots when both plot variables are numeric, and FALSE is in effect for options DISCRETEX=, DISCRETEY=, XBINAXIS=, and YBINAXIS. As long as one of these plots is present, you can also add FRINGEPLOTPLOT, LINEPLOTPLOT, MODELBAND, REFERENCENAMELINE, DROPLINE,
DISCRETELEGEND, CONTINUOUSLEGEND, and text-based statements such as ENTRY.

This layout has only two independent axes from a data standpoint, X and Y. If any contained plot uses an X=X2 or Y=Y2 option, then the option is ignored and the data is mapped to the X or Y axis. However, the X2 and Y2 axes can be displayed using the DISPLAY2= suboption of the XAXISOPTS= and YAXISOPTS= options.

Interaction: When nested within another layout type, the OVERLAYEQUATED layout defines the graph display for one cell of the parent layout. A separate OVERLAYEQUATED layout is specified for each cell.

Tips: By default, the first plot specified in the LAYOUT OVERLAYEQUATED block defines the axis characteristics. If you want another plot to define the axis characteristics, specify PRIMARY=TRUE in the statement for that plot.

Overlaid plots are stacked in the order in which they are specified. The last plot specified is placed on top. When plots are stacked, a plots’s graphical data might obscure the graphical data of the plots beneath it. In that case, you can change the plot order or use transparency to make all of the graphical data visible.

Syntax

LAYOUT OVERLAYEQUATED <options>;
  GTL-statements;
ENDLAYOUT;

Summary of Optional Arguments

Appearance options

BACKGROUNDCOLOR=style-reference | color
  specifies the color of the layout background.

BORDER=TRUE | FALSE
  specifies whether a border is drawn around the layout.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the border line around the layout.

CYCLEATTRS=TRUE | FALSE
  specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.

OPAQUE=TRUE | FALSE
  specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

OUTERPAD=AUTO | dimension | (pad-options)
  specifies the amount of extra space to add outside the layout border.

PAD=dimension | (pad-options)
  specifies the amount of extra space that is added inside the layout border.

WALLCOLOR=style-reference | color
  specifies the fill color of the plot wall area.

WALLDISPLAY=STANDARD | ALL | NONE | (display-options)
  specifies whether the plot’s wall and wall outline are displayed.

Axes options

COMMONAXISOPTS=(common-axis-options)
  specifies one or more options to apply to both the X and Y equated axes.
EQUATETYPE=FIT | SQUARE | SQUAREDATA | EQUATE
    specifies how to draw the axis area.
XAXISOPTS=(axis-options)
    specifies one or more X-axis options.
YAXISOPTS=(axis-options)
    specifies one or more Y-axis options.

Optional Arguments
BACKGROUNDCOLOR=style-reference | color
    specifies the color of the layout background.

    style-reference
    specifies a style reference in the form style-element:style-attribute.
    Only the style attribute named COLOR or CONTRASTCOLOR is used.

    color
    specifies a color by name.

    Default
    The GraphBackground:Color style reference.

    Interaction
    OPAQUE=TRUE must be in effect in order for the color to be seen.
    By default, OPAQUE=FALSE.

BORDER=TRUE | FALSE
    specifies whether a border is drawn around the layout.

    Default
    FALSE

    Interaction
    If this option is set to FALSE, then the BORDERATTRS= option is ignored.

    See
    “boolean ” on page 1479 for other Boolean values that you can use.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
    specifies the attributes of the border line around the layout.

    Default
    The GraphBorderLines style element.

    Interaction
    BORDER= TRUE must be set for this option to have any effect.

    See
    “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

    “Line Options” on page 1522 for available line-options.

COMMONAXISOPTS=(common-axis-options)
    specifies one or more options to apply to both the X and Y equated axes.

    Requirements
    Axis options must be enclosed in parentheses.

    Each option must be specified as a name = value pair and must be separated by a space.

    See
    “Options That Apply to Both Equated Axes” on page 1127 for a list of options.
CYCLEATR=TRUE | FALSE
specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.

FALSE
does not cycle the default visual attributes of multiple plots.
For example, if you overlay three series plots, then each series line has the same default visual properties.

TRUE
attempts to use the GraphData1–GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others).
Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

Default FALSE

See “Rotating Visual Attributes for Each Plot in an Overlay” on page 197 “boolean ” on page 1479 for other Boolean values that you can use.

EQUATETYPE=FIT | SQUARE | SQUAREDATA | EQUATE
specifies how to draw the axis area.

FIT
specifies that the X and Y axes have equal increments between tick values.
The data ranges of both axes are compared to establish a common increment size. The axes might be of different lengths and have a different number of tick marks. Each axis represents its own data range. One axis might be extended to use available space in the plot area. If a TICKVALUELIST= or TICKVALUESEQUENCE= axis option is used on COMMONAXISOPTS=, then it is ignored.

SQUARE
specifies that both the X and Y axes have the same length and the same major tick values.
The axis length and tick values are chosen so that the minimum and maximum of both X and Y appear in the range of values appearing on both axes.

**SQUAREDATA**

specifies that both the X and Y axes have the same data range, but they can have different tick values.

A UNION of the data ranges does not occur in this case. For example, if the X-axis values are 20 to 40 (range of 20) and the Y-axis values are 200 to 260 (range of 60), then both axes have a range of 60 units, but the X axis can have tick values 0, 20, 40, and 60, and the Y axis can have tick values 200, 220, 240, and 260.

**EQUATE**

same as FIT except that neither axis is extended to use available space in the plot area.

Default: FIT

**OPAQUE=TRUE | FALSE**

specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default: FALSE

Interaction: When this option is set to FALSE, the background color is not used.

See: "boolean " on page 1479 for other Boolean values that you can use.

**OUTERPAD=AUTO | dimension | (pad-options)**

specifies the amount of extra space to add outside the layout border.

**AUTO**

specifies that the default outside padding for this component is used.

**dimension**

specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

**(pad-options)**

a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can contain one or more of the following pad options:

**LEFT=dimension**

specifies the amount of extra space to add to the left side.

**RIGHT=dimension**

specifies the amount of extra space to add to the right side.

**TOP=dimension**

specifies the amount of extra space to add to the top.

**BOTTOM=dimension**

specifies the amount of extra space to add to the bottom.

**Note** Sides that are not assigned padding are padded with the default amount.

**Tip** Use pad-options to create non-uniform padding.
PAD=*dimension* | *(pad-options)*
specifies the amount of extra space that is added inside the layout border.

*dimension*
specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

*(pad-options)*
a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

LEFT=*dimension*
specifies the amount of extra space added to the left side.

Default: 0

RIGHT=*dimension*
specifies the amount of extra space added to the right side.

Default: 0

TOP=*dimension*
specifies the amount of extra space added to the top.

Default: 0

BOTTOM=*dimension*
specifies the amount of extra space added to the bottom.

Default: 0

Note: Sides that are not assigned padding are padded with the default amount.

Tip: Use *pad-options* to create non-uniform padding.

WALLCOLOR=*style-reference* | *color*
specifies the fill color of the plot wall area.

*style-reference*
specifies a style reference in the form *style-element:* *style-attribute*.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

*color*
specifies a color by name.
### WALLDISPLAY

**WALLDISPLAY=STANDARD | ALL | NONE | (display-options)**

specifies whether the plot’s wall and wall outline are displayed.

**STANDARD**
- displays a filled wall.
  - The setting of the FRAMEBORDER= attribute of the GraphWalls style element determines whether the wall outline is displayed.

**ALL**
- displays a filled, outlined wall.

**NONE**
- displays no wall and no wall outline.

**display-options**
- specifies a space-separated list of one or more display options, enclosed in parentheses.
  - The list can contain one or more of the following options:
    - **OUTLINE** displays the wall outline.
    - **FILL** displays a filled wall area.

**Default** STANDARD

**Tips** Use the WALLCOLOR= option to control the fill color of the wall.

The appearance attributes of the wall fill and outline are set by the GraphWalls style element.

### XAXISOPTS=(axis-options)

specifies one or more X-axis options.

**Requirements**
- Axis options must be enclosed in parentheses and separated by spaces.
  - Each option must be specified as a name = value pair, and each pair must be separated by a space.

**See**
- "Options That Apply Separately to an X or Y Equated Axis" on page 1133 for a list of options.

### YAXISOPTS=(axis-options)

specifies one or more Y-axis options.

**Requirements**
- Axis options must be enclosed in parentheses and separated by spaces.
  - Each option must be specified as a name = value pair, and each pair must be separated by a space.

**See**
- "Options That Apply Separately to an X or Y Equated Axis" on page 1133 for a list of options.
Details

The LAYOUT OVERLAYEQUATED statement is similar to the LAYOUT OVERLAY statement: it builds a composite using one or more GTL-statements. Similar to a LAYOUT OVERLAY, you can specify one or more 2-D plots within the layout, provided all plots can share the same type of axes. (Additional restrictions are discussed in a moment.) You can also specify one or more insets.

As an overlay-type layout, OVERLAYEQUATED has the same behavioral characteristics as an OVERLAY layout. It uses the same general logic for rendering the composite (see “LAYOUT OVERLAY Statement” on page 149 for details), and its default axis characteristics are generally determined by the first specified plot, unless you use PRIMARY=TRUE on an alternative plot statement (see “When Plots Share Data and a Common Axis” on page 979).

OVERLAYEQUATED differs from OVERLAY in several ways. With OVERLAYEQUATED,

- The axis type for both X and Y axes is always linear. Thus, plot types that have discrete or binned axes cannot be used within this layout. This includes BOXPLOT, BOXPLOTPARM, BARCHARTPARM, HISTOGRAM, and HISTOGRAMPARM. It also includes HEATMAP and HEATMAPPARM when one or both axes are binned.

- For equal data intervals on both axes, the display distance is the same. For example, an interval of 2 on the X axis maps to the same display distance as an interval of 2 on the Y axis.

- The aspect ratio of the plot display equals the aspect ratio of the plot data. In other words, a 45-degree slope in data is represented by a 45-degree slope in the display. The EQUATETYPE= option determines how the axes are drawn.

The following figure illustrates how a series plot might map differently when specified in an OVERLAYEQUATED layout versus an OVERLAY layout:

From a data standpoint, this layout has only two independent axes, X and Y. If any plots within the layout block use an XAXIS=X2 or YAXIS=Y2 option, then the option is ignored and the data are mapped to the X or Y axis. To display X2 and Y2 axes, use the DISPLAYSECONDARY= suboption of the XAXISOPTS= and YAXISOPTS= options.
If an OVERLAYEQUATED statement is nested in a LATTICE layout, then some of the LATTICE’s alignment and external axis features are not supported on the OVERLAYEQUATED layout.

Example: LAYOUT OVERLAYEQUATED Statement

The following graph was generated by the "Example Program" on page 165:

Example Program

```plaintext
proc template;
   define statgraph layoutoverlayequated;
   begingraph;
      entrytitle "Gas Mileage for GMC Models";
      layout overlayequated / equatetype=fit;
      referenceline y=16.2 / curvelabel="City Average for Trucks/SUVs"
      curvelabellocation=inside curvelabelattrs=GraphReference;
      referenceline x=20.6 / curvelabel="Highway Average for Trucks/SUVs"
      curvelabellocation=inside curvelabelattrs=GraphReference;
      scatterplot x=mpg_highway y=mpg_city / datalabel=model;
      endlayout;
   endgraph;
end;
run;
proc sgrender data=sashelp.cars
```
template=layoutoverlayequated;
where make="GMC";
run;

LAYOUT OVERLAY3D Statement

Builds a 3-D composite from one or more GTL-statements. The composite could be an entire graph. Or, if this layout is nested in a GRIDDED or LATTICE layout, then the composite typically provides contents for one cell in the parent layout.

Restriction: You can add one or more 3-D plots to the graph area that the LAYOUT OVERLAY3D statement creates, but all of the graphs will share the same set of axes.

Tips: By default, the first plot specified in the LAYOUT OVERLAY3D block defines the axis characteristics. If you want another plot to define the axis characteristics, specify PRIMARY=TRUE in the statement for that plot.
Overlaid plots are stacked in the order in which they are specified. The last plot specified is placed on top. When plots are stacked, a plot’s graphical data might obscure the graphical data of the plots beneath it. In that case, you can change the plot order or use transparency to make all of the graphical data visible.
The LAYOUT OVERLAY3D block is primarily used to specify 3-D plot statements, but you can also specify other statements such as CONTINUOUSLEGEND.

Syntax

LAYOUT OVERLAY3D <options>;
  GTL-statements;
ENDLAYOUT;

Summary of Optional Arguments

Appearance options

  BACKGROUNDCOLOR=style-reference | color
  specifies the color of the layout background.

  BORDER=TRUE | FALSE
  specifies whether a border is drawn around the layout.

  BORDERATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the border line around the layout.

  CUBE=TRUE | FALSE
  specifies whether the layout displays the lines that indicate the complete bounding cube of the axis planes.

  CYCLEATTRS=TRUE | FALSE
  specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.

  OPAQUE=TRUE | FALSE
  specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

  OUTERPAD=AUTO | dimension | (pad-options)
  specifies the amount of extra space to add outside the layout border.

  PAD=dimension | (pad-options)
specifies the amount of extra space that is added inside the layout border.

**ROTATE**=*number*

- specifies the angle of rotation.

**TILT**=*number*

- specifies the angle of tilt in degrees.

**WALLCOLOR**=*style-reference | color*

- specifies the fill color of the plot wall area.

**WALLDISPLAY**=*STANDARD | ALL | NONE | (display-options)*

- specifies whether the plot’s wall and wall outline are displayed.

**ZOOM**=*positive-number*

- specifies a zoom factor.

### Axes options

**XAXISOPTS**=(*axis-options*)

- specifies one or more X-axis options.

**YAXISOPTS**=(*axis-options*)

- specifies one or more Y-axis options.

**ZAXISOPTS**=(*axis-options*)

- specifies one or more Z-axis options.

### Optional Arguments

**BACKGROUND COLOR**=*style-reference | color*

- specifies the color of the layout background.

- *style-reference*
  
  - specifies a style reference in the form *style-element:style-attribute*.
  
  Only the style attribute named COLOR or CONTRASTCOLOR is used.

- *color*
  
  - specifies a color by name.

See  color on page 1480

**Default**  The GraphBackground:Color style reference.

**Interaction**  OPAQUE=TRUE must be in effect in order for the color to be seen. By default, OPAQUE=FALSE.

**BORDER**=*TRUE | FALSE*

- specifies whether a border is drawn around the layout.

**Default**  FALSE

**Interaction**  If this option is set to FALSE, then the BORDERATTRS= option is ignored.

See  “boolean “ on page 1479 for other Boolean values that you can use.

**BORDERATTRS**=*style-element | style-element (line-options) | (line-options)*

- specifies the attributes of the border line around the layout.

**Default**  The GraphBorderLines style element.

**Interaction**  BORDER= TRUE must be set for this option to have any effect.
See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

CUBE=TRUE | FALSE
specifies whether the layout displays the lines that indicate the complete bounding cube of the axis planes.

Default TRUE

Note The cube lines are displayed independently of the wall borders and axis lines. Because some cube lines coincide with wall borders and axis lines, it might appear that turning off wall borders or axis lines has no effect when CUBE=TRUE.

Tip The color, thickness, and pattern of the cube lines are determined by the GraphAxisLines style element.

See “boolean ” on page 1479 for other Boolean values that you can use.

CYCLECHARS=TRUE | FALSE
specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.

FALSE
does not cycle the default visual attributes of multiple plots.
For example, if you overlay three series plots, then each series line has the same default visual properties.

TRUE
attempts to use the GraphData1–GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others).
Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

Default FALSE

See “Rotating Visual Attributes for Each Plot in an Overlay” on page 197
“boolean ” on page 1479 for other Boolean values that you can use.

OPAQUE=TRUE | FALSE
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

Default FALSE

Interaction When this option is set to FALSE, the background color is not used.

See “boolean ” on page 1479 for other Boolean values that you can use.

OUTERPAD=AUTO | dimension | (pad-options)
specifies the amount of extra space to add outside the layout border.

AUTO
specifies that the default outside padding for this component is used.
**dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

**(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can contain one or more of the following pad options:

LEFT=**dimension**
specifies the amount of extra space to add to the left side.

RIGHT=**dimension**
specifies the amount of extra space to add to the right side.

TOP=**dimension**
specifies the amount of extra space to add to the top.

BOTTOM=**dimension**
specifies the amount of extra space to add to the bottom.

**Note** Sides that are not assigned padding are padded with the default amount.

**Tip** Use **pad-options** to create non-uniform padding.

**Default** AUTO

**Note** The default units for **dimension** are pixels.

**See** [“**dimension**” on page 1480]

**PAD=** **dimension** | **(pad-options)**
specifies the amount of extra space that is added inside the layout border.

**dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.

**(pad-options)**
a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

LEFT=**dimension**
specifies the amount of extra space added to the left side.

**Default** 0

RIGHT=**dimension**
specifies the amount of extra space added to the right side.

**Default** 0

TOP=**dimension**
specifies the amount of extra space added to the top.

**Default** 0
BOTTOM=\textit{dimension}  
specifies the amount of extra space added to the bottom.  

\textbf{Default} \hspace{1em} 0  

\textbf{Note} \hspace{1em} Sides that are not assigned padding are padded with the default amount.  

\textbf{Tip} \hspace{1em} Use \textit{pad-options} to create non-uniform padding.  

\textbf{Note} \hspace{1em} The default units for \textit{dimension} are pixels.  

\textbf{See} \hspace{1em} "\textit{dimension} on page 1480"

\textbf{ROTATE=}\textit{number}  
specifies the angle of rotation. Rotation is measured in a clockwise direction about a virtual axis parallel to the Z axis (vertical) and passing through the center of the bounding cube. A counterclockwise rotation can be specified with a negative value.  

\textbf{Default} \hspace{1em} 54  

\textbf{TILT=}\textit{number}  
specifies the angle of tilt in degrees. Tilt is measured in a clockwise direction about a virtual axis parallel to the X axis (vertical) and passing through the center of the bounding cube. A counterclockwise rotation can be specified with a negative value.  

\textbf{Default} \hspace{1em} 20  

\textbf{WALLCOLOR=}\textit{style-reference} | \textit{color}  
specifies the fill color of the plot wall area.  

\textit{style-reference}  
specifies a style reference in the form \textit{style-element}:\textit{style-attribute}.  

Only the style- attribute named COLOR or CONTRASTCOLOR is used.  

\textit{color}  
specifies a color by name.  

\textbf{See} \hspace{1em} color on page 1480  

\textbf{Default} \hspace{1em} The GraphWalls:Color style reference.  

\textbf{Interaction} \hspace{1em} This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).  

\textbf{WALLDISPLAY=}\textbf{STANDARD} | \textbf{ALL} | \textbf{NONE} | (display-options)  
specifies whether the plot’s wall and wall outline are displayed.  

\textbf{STANDARD}  
displays a filled wall.  

The setting of the FRAMEBORDER= attribute of the GraphWalls style element determines whether the wall outline is displayed.  

\textbf{ALL}  
displays a filled, outlined wall.
NONE
  displays no wall and no wall outline.

(display-options)
  specifies a space-separated list of one or more display options, enclosed in parentheses.
  The list can contain one or more of the following options:

  OUTLINE  displays the wall outline.
  FILL      displays a filled wall area.

Default  STANDARD

Tips
  Use the WALLCOLOR= option to control the fill color of the wall.
  The appearance attributes of the wall fill and outline are set by the GraphWalls style element.

See  the CUBE= option.

XAXISOPTS=(axis-options)
  specifies one or more X-axis options.

Requirements Axis options must be enclosed in parentheses and separated by spaces.
  Each option must be specified as a name = value pair, and each pair must be separated by a space.

See  "Axis Options for LAYOUT OVERLAY3D” on page 1051 for a list of axis options.

YAXISOPTS=(axis-options)
  specifies one or more Y-axis options.

Requirements Axis options must be enclosed in parentheses and separated by spaces.
  Each option must be specified as a name = value pair, and each pair must be separated by a space.

See  "Axis Options for LAYOUT OVERLAY3D” on page 1051 for a list of axis options.

ZAXISOPTS=(axis-options)
  specifies one or more Z-axis options.

Requirements Axis options must be enclosed in parentheses and separated by spaces.
  Each option must be specified as a name = value pair, and each pair must be separated by a space.

See  "Axis Options for LAYOUT OVERLAY3D” on page 1051 for a list of axis options.
ZOOM=\textit{positive-number}

specifies a zoom factor. Factors greater than 1 move closer to the bounding cube, less than 1 move farther away

\textbf{Default} \hspace{1cm} 1

\section*{Details}

The \texttt{LAYOUT OVERLAY3D} statement builds a 3-D composite using one or more \texttt{GTL-statements}. You can specify one or more 3-D plots within the layout, provided all plots can share the same type of axes. You can also specify “annotations” (for example, with one or more \texttt{ENTRY} statements or \texttt{LAYOUT GRIDDED} statements). However, annotations in the \texttt{OVERLAY3D} layout are more likely to collide with other graphics features than are annotations in other overlay-type layouts.

As an overlay-type layout, \texttt{OVERLAY3D} has the same behavioral characteristics as an \texttt{OVERLAY} layout. It uses the same general logic for rendering the composite (see “\texttt{LAYOUT OVERLAY} Statement” on page 149 for details), and its default axis characteristics are generally determined by the first specified plot, unless you use \texttt{PRIMARY=TRUE} on another plot statement (see “When Plots Share Data and a Common Axis” on page 979).

Within an \texttt{OVERLAY3D} layout, a graph’s bounding cube can be tilted, rotated, and zoomed to provide a different viewpoint. By default, the outline of the bounding cube is displayed and the viewing rotation angle is 57 degrees, the tilt angle is 20 degrees, and the zoom factor is 1. See the \texttt{CUBE=}, \texttt{ROTATE=} , \texttt{TILT=} , and \texttt{ZOOM=} options for information about how to change the viewpoint.

\section*{Example: \texttt{LAYOUT OVERLAY3D} Statement}

The following graph was generated by the “Example Program” on page 173:
Example Program

```sas
proc template;
  define statgraph layoutoverlay3d;
  begingraph;
    entrytitle "Density Plot of Height and Weight";
    layout overlay3d / tilt=10 rotate=54
      walldisplay=none cube=false;
      surfaceplotparm x=height y=weight z=density /
        surfacecolorgradient=density;
    endlayout;
  endgraph;
end;
run;
proc sgrender data=sashelp.gridded template=layoutoverlay3d;
run;
```

LAYOUT PROTOTYPE Statement

Builds a composite from one or more plot-statements. The composite is used as a prototype or "rubber stamp" that repeats in each cell of a parent DATALATTICE or DATAPANEL layout.

Restrictions:

You can specify only one LAYOUT PROTOTYPE block in a LAYOUT DATAPANEL or LAYOUT DATALATTICE block. If you specify more than one, then only the last prototype block specified is honored. The remaining prototype blocks are ignored.

Only the following plots can be used in a LAYOUT PROTOTYPE block: BANDPLOT, BARCHART, BARCHARTPARM, BLOCKPLOT, BOXPLOTPARM, COUNTOURPLOTPARM, DROPLINE, ELLIPSEPARM, FRINGEPLOT, HEATMAPPARM, HISTOGRAMPARM, LINECHART, LINEPARM, NEEDLEPLOT, REFERENCELINE, SCATTERPLOT, SERIESPLOT, STEPPLOT, and VECTORPLOT.

SCATTERPLOTMATRIX plots, 3-D plots, and region plots such as PIECHART or MOSAICPLOTPARM cannot be used in the LAYOUT PROTOTYPE block.

ENTRY, DISCRETELEGEND, and CONTINUOUSLEGEND statements cannot be used in the LAYOUT PROTOTYPE block.

A plot statement cannot be used if it contains a column defined with an EVAL expression.

You can add one or more two-dimensional plots and one-dimensional plots to the graph area that the LAYOUT PROTOTYPE statement creates, provided all of the graphs can share the same axis type.

If you include a plot statement with a CURVELABEL= option (such as SERIESPLOT), then only CURVELABELLOCATION=INSIDE is supported.

If you include a plot statement that supports a CLIP= option (such as LINEPARM or ELLIPSEPARM), then the CLIP value is always set to TRUE.

Requirement: The LAYOUT PROTOTYPE statement must be nested in a LAYOUT DATAPANEL or LAYOUT DATALATTICE block.

Note: Nesting an INNERMARGIN block in the LAYOUT PROTOTYPE statement is valid in SAS 9.4M1 and later releases.

See:

"LAYOUT DATAPANEL Statement" on page 74
"LAYOUT DATALATTICE Statement" on page 47
Syntax

LAYOUT PROTOTYPE </option(s)>;
plot-statements;
<INNERMARGIN </options(s)>;
  block-plot-statement(s); | axis-table statement(s);
ENDINNERMARGIN>
<... more-innermargin-blocks ...> 
ENDLAYOUT;

Optional Arguments

ASPECTRATIO=AUTO | positive-number
specifies the aspect ratio of the prototype cell. The ratio is expressed as a positive decimal fraction representing wall-height divided by wall-width. For example, 0.75 is a 3/4 aspect ratio and 1.0 is a square aspect ratio.

Default AUTO. The prototype cell is sized to the maximum area that can fill the available space inside the layout cell.

Note If AUTO is not used for the aspect ratio, then the entire DATALATTICE or DATAPANEL grid is affected and changes shape.

CYCLEATTRS=TRUE | FALSE
specifies whether the default visual attributes of markers, lines, and fills in nested plot statements automatically change from plot to plot.

FALSE
does not cycle the default visual attributes of multiple plots.

For example, if you overlay three series plots, then each series line has the same default visual properties.

TRUE
attempts to use the GraphData1–GraphDataN style elements to assign different visual properties to applicable plots (scatter plots and series plots and others).

Some plots in the layout do not participate in the cycling (for example, reference lines and drop lines).

Default FALSE

See “Rotating Visual Attributes for Each Plot in an Overlay” on page 197 “boolean ” on page 1479 for other Boolean values that you can use.

WALLCOLOR=style-reference | color
specifies the fill color of the plot wall area.

style-reference
specifies a style reference in the form style-element:style-attribute.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

color
specifies a color by name.

See color on page 1480
### Default
The GraphWalls:Color style reference.

### Interaction
This option is ignored if WALLDISPLAY=NONE or
WALLDISPLAY=(OUTLINE).

### WALLDISPLAY=STANDARD | ALL | NONE | (display-options)
specifies whether the plot’s wall and wall outline are displayed.

- **STANDARD**  
  displays a filled wall.
  
  The setting of the FRAMEBORDER= attribute of the GraphWalls style element determines whether the wall outline is displayed.

- **ALL**  
  displays a filled, outlined wall.

- **NONE**  
  displays no wall and no wall outline.

- **(display-options)**  
  specifies a space-separated list of one or more display options, enclosed in parentheses.
  
  The list can contain one or more of the following options:

  - **OUTLINE**  
    displays the wall outline.

  - **FILL**  
    displays a filled wall area.

### Default
**STANDARD**

### Tips
Use the WALLCOLOR= option to control the fill color of the wall.

- The appearance attributes of the wall fill and outline are set by the GraphWalls style element.

- When the wall outline is suppressed, adjacent lines such as axis lines and cell-header borders are still displayed. To suppress the axis lines, use the appropriate display option for the axes. The cell-header borders cannot be suppressed.

### Details
The LAYOUT PROTOTYPE statement defines a plot prototype or “rubber stamp” that repeats automatically. It assembles the results of nested GTL statements into a single axis area. The plots are drawn in the order in which they are specified. The results of the last statement are placed on top.

The plot-statements determine the graphical content of the cells in the parent layout, based on the subsetting of the specified classification variables. For an example, see “LAYOUT DATALATTICE Statement” on page 47 or “LAYOUT DATAPANEL Statement” on page 74.

A PROTOTYPE layout is essentially a restricted OVERLAY layout with the same general rules for overlaying plots. The main difference is that there are no axis options available on the LAYOUT PROTOTYPE statement. Axis properties are set with the ROWAXISOPTS= and COLUMNAXISOPTS= options of the parent DATAPANEL or DATALATTICE statement.
In SAS 9.4 and later releases, a PROTOTYPE layout can also contain an inner margin, which is a nested region at the top or bottom of the PROTOTYPE container. One or more inner margin plots can be specified, and each is specified within an INNERMARGIN block. Within the INNERMARGIN block, only one-dimensional plot statements such as BLOCKPLOT and AXISTABLE can be specified. See “INNERMARGIN Statement” on page 179.

### LAYOUT REGION Statement

Creates the drawing area for a plot that does not use axes.

**Restrictions:**
- A LAYOUT REGION block cannot contain more than one plot.
- A LAYOUT REGION block can contain a PIECHART or MOSAICPLOTPARM plot only.

**Syntax**

```
LAYOUT REGION </options> ;
GTL-statements;
ENDLAYOUT;
```

**Optional Arguments**

**BACKGROUNDCOLOR=**

`style-reference | color`

specifies the color of the layout background.

- **style-reference**
  - specifies a style reference in the form `style-element:style-attribute`.
  - Only the style attribute named COLOR or CONTRASTCOLOR is used.
- **color**
  - specifies a color by name.

**Default**

The GraphBackground:Color style reference.

**Interaction**

`OPAQUE=TRUE` must be in effect in order for the color to be seen. By default, `OPAQUE=FALSE`.

**BORDER=**

`TRUE | FALSE`

specifies whether a border is drawn around the layout.

- **Default**
  - FALSE
- **Interaction**
  - If this option is set to FALSE, then the BORDERATTRS= option is ignored.

**BORDERATTRS=**

`style-element | style-element (line-options) | (line-options)`

specifies the attributes of the border line around the layout.

- **Default**
  - The GraphBorderLines style element.
Interacion  **BORDER=** TRUE must be set for this option to have any effect.

<table>
<thead>
<tr>
<th>See</th>
<th>“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“Line Options” on page 1522 for available line-options.</td>
</tr>
</tbody>
</table>

**OPAQUE=TRUE | FALSE**
specifies whether the layout background is opaque (TRUE) or transparent (FALSE).

<table>
<thead>
<tr>
<th>Default</th>
<th>FALSE</th>
</tr>
</thead>
</table>

Interaction  When this option is set to FALSE, the background color is not used.

| See | “boolean” on page 1479 for other Boolean values that you can use. |

**OUTERPAD=AUTO | dimension | (pad-options)**
specifies the amount of extra space to add outside the layout border.

- **AUTO**
specifies that the default outside padding for this component is used.

- **dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the layout border.

- **(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

  The list can contain one or more of the following pad options:

  - **LEFT=** *dimension*
specifies the amount of extra space to add to the left side.

  - **RIGHT=** *dimension*
specifies the amount of extra space to add to the right side.

  - **TOP=** *dimension*
specifies the amount of extra space to add to the top.

  - **BOTTOM=** *dimension*
specifies the amount of extra space to add to the bottom.

  **Note** Sides that are not assigned padding are padded with the default amount.

  **Tip** Use pad-options to create non-uniform padding.

<table>
<thead>
<tr>
<th>Default</th>
<th>AUTO</th>
</tr>
</thead>
</table>

| Note | The default units for *dimension* are pixels. |

| See | “dimension” on page 1480 |

**PAD=** *dimension | (pad-options)**
specifies the amount of extra space that is added inside the layout border.

- **dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.
a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

**LEFT=** *dimension*

specifies the amount of extra space added to the left side.

Default 0

**RIGHT=** *dimension*

specifies the amount of extra space added to the right side.

Default 0

**TOP=** *dimension*

specifies the amount of extra space added to the top.

Default 0

**BOTTOM=** *dimension*

specifies the amount of extra space added to the bottom.

Default 0

---

**Note** Sides that are not assigned padding are padded with the default amount.

**Tip** Use **pad-options** to create non-uniform padding.

---

**Note** The default units for *dimension* are pixels.

**See** “*dimension*” on page 1480

---

**Details**

The REGION layout provides a container for plots that do not use axes. Within the LAYOUT REGION block, you can specify a single plot statement of a type that never uses axes, such as a PIECHART or MOSAICPLOTPARM. If multiple plot statements are specified, then only the first one is honored. You can also specify one or more insets, such as nested layout statements (for example, LAYOUT GRIDDED), ENTRY statements, and legend statements (CONTINUOUSLEGEND or DISCRETELEGEND). For example, you could specify a PIECHART statement with a DISCRETELEGEND statement and an ENTRY statement. You can also nest one or more layout blocks within the REGION layout. For example, you could nest a LAYOUT GRIDDED statement that creates a small table of text.

When nested within another layout type, such as a GRIDDED or LATTICE layout, the REGION layout defines the graphical display for one cell of the parent layout. A separate REGION layout is specified for each cell.

**Example: LAYOUT REGION Statement**

The following graph was generated by the “Example Program” on page 179:
Example Program

```plaintext
proc template;
  define statgraph layoutregion;
  begingraph;
    entrytitle "Average Weight by Age";
    layout region;
      piechart category=age response=weight /
        stat=mean name="p"
        datalabelcontent=(response) datalabellocation=outside;
      discretelegend "p" / title="Age" across=2
        border=true halign=right valign=top;
    endlayout;
  endgraph;
end;

proc sgrender data=sashelp.class template=layoutregion;
run;
```

INNERMARGIN Statement

Provides a nested region in a LAYOUT OVERLAY or LAYOUT PROTOTYPE container in which a block plot or axis table can be placed.

Restriction: This statement is valid in LAYOUT OVERLAY and LAYOUT PROTOTYPE blocks only.

Notes: Two or more INNERMARGIN blocks that have the same alignment are stacked.
Multiple statements within an INNERMARGIN block are stacked.
For an X axis, the offsets on each end of the Y axis are increased to make room for the inner margin plots. For a Y axis, the offsets on each end of the X axis are increased to make room for the inner margin plots.

Syntax

```
INNERMARGIN < /options> ;
    block-plot-statement(s); | axis-table statement(s);
ENDINNERMARGIN;
```

Optional Arguments

```
ALIGN=TOP | BOTTOM | LEFT | RIGHT
```

specifies the alignment of the inner margin.

**Default**

BOTTOM

**Restrictions**

For a block plot, only TOP and BOTTOM are valid. LEFT and RIGHT are ignored.

For an axis table, LEFT and RIGHT can be used for a Y or Y2 axis.

Multiple statements within an INNERMARGIN block are stacked.

**Note**

For an inner margin with ALIGN=TOP or ALIGN=BOTTOM, the offsets on each end of the Y axis are increased to reserve space for the inner margin plots. For an inner margin with ALIGN=LEFT or ALIGN=RIGHT, the offsets on each end of the X axis are increased to reserve space for the inner margin plots.

**BACKGROUNDCOLOR=style-reference | color**

specifies the color of the inner margin background.

```
style-reference
```

specifies a style reference in the form style-element:style-attribute. Only the style attributes named COLOR or CONTRASTCOLOR are used.

```
color
```

specifies a color by name.

See color on page 1480
### INNERMARGIN Statement

<table>
<thead>
<tr>
<th>Default</th>
<th>Interaction</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>The graph wall color. (See WALLCOLOR=.)</td>
<td>For this option to have any effect, the OPAQUE= option must be set to TRUE.</td>
<td>The inner margin background is set to the wall color even when WALLDISPLAY= NONE.</td>
</tr>
</tbody>
</table>

### GUTTER=dimension

specifies the gap between stacked items in the inner margin.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

<table>
<thead>
<tr>
<th>Default</th>
<th>Requirement</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The inner margin must contain two or more stacked items for this option to have any effect.</td>
<td>The default units for dimension are pixels.</td>
</tr>
</tbody>
</table>

**See** "dimension" on page 1480

### OPAQUE=TRUE | FALSE

specifies whether the inner margin's background is opaque.

**TRUE**
specifies that the background is opaque.

**FALSE**
specifies that the background is transparent.

<table>
<thead>
<tr>
<th>Default</th>
<th>Interaction</th>
<th>Tip</th>
<th>See</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>When this option is FALSE, the BACKGROUNDCOLOR= option is ignored.</td>
<td>To prevent axis color bars and grid lines from passing through the axis table, set OPAQUE=TRUE.</td>
<td>&quot;boolean &quot; on page 1479 for other Boolean values that you can use.</td>
</tr>
</tbody>
</table>

### PAD=dimension | (pad-options)

specifies the amount of extra space that is added inside the inner-margin border.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

<table>
<thead>
<tr>
<th>dimension</th>
<th>(pad-options)</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies a dimension to use for the extra space at the left, right, top, and bottom of the border.</td>
<td>a space-separated list of one or more name-value-pair pad options enclosed in parentheses.</td>
</tr>
</tbody>
</table>

The list can include one or more of the following pad options:

**LEFT=dimension**
specifies the amount of extra space added to the left side.
RIGHT=dimension
specifies the amount of extra space added to the right side.

Default 0

TOP=dimension
specifies the amount of extra space added to the top.

Default 5 px for the first inner margin adjacent to the bottom of the plot area. Otherwise, 0.

BOTTOM=dimension
specifies the amount of extra space added to the bottom.

Default 5 px for the first inner margin adjacent to the top of the plot area. Otherwise, 0.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use pad-options to create non-uniform padding.

Note The default units for dimension are pixels.

See “dimension” on page 1480

SEPARATOR=TRUE | FALSE
specifies whether a separating line is drawn between the inner margin and the rest of the layout content.

Note: This option is valid starting in SAS 9.4M1.

Default FALSE

Tip Use the SEPARATORATTRS= option to specify the attributes of the separating line.

See “Example: Overlay with an Inner Margin Plot” on page 183

“boolean ” on page 1479 for other Boolean values that you can use.

SEPARATORATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the inner margin separating line.

Note: This option is valid starting in SAS 9.4M1.

Default The graphAxisLines style element

Interaction This option is ignored when SEPARATOR=FALSE.
Details

An inner margin is a nested region in an OVERLAY container. You can specify one or more inner margin plots. Specify each plot within an INNERMARGIN block. Within an INNERMARGIN block, you can specify only the BLOCKPLOT and AXISTABLE statements. See “Example: Overlay with an Inner Margin Plot” on page 183.

Example: Overlay with an Inner Margin Plot

Example Overview

This example shows how to place a plot in an inner margin of an OVERLAY layout. It creates a graph that shows monthly total sales for a specific year. A LINECHART statement is used to draw the plot. The months are shown along the category axis, and the total sales values are shown along the response axis. The Sashelp.Prdsale data set is used as the data source. The tick marks on the category axis are positioned between the midpoints to align with the beginning of each month.

A BLOCKPLOT statement in an INNERMARGIN block is used to display the quarters above the category axis. The INNERMARGIN statement uses the default alignment, so the inner margin is positioned at the bottom of the layout container, beneath the line chart. Alternate shading is used in the block plot to show the block boundaries. Because the tick marks are positioned between the midpoints, they align with the block boundaries. The SEPARATOR= and SEPARATORATTRS= options are used in the INNERMARGIN statement to specify a dark-red, two-pixel-wide separator line between the inner margin and the rest of the graph.

Note: The INNERMARGIN statement SEPARATOR= and SEPARATORATTRS= options are valid in SAS 9.4M1 and later releases.

Example Output

Here is the output for this example.
Example Program

Here is the SAS code.

```sas
/* Create a format for the quarters */
proc format;
value quartername 1="Quarter 1" 2="Quarter 2"
3="Quarter 3" 4="Quarter 4";
run;

/* Define the graph template */
proc template;
define statgraph innermargin;
dynamic year;
begin graph / subpixel=on;
entrytitle "Total Sales in " year;
layout overlay /
  xaxisopts=(type=discrete discreteopts=(ticktype=inbetween));
innermargin /
  separator=true
  separatorattrs=(color=darkred thickness=2px);
blockplot x=month block=quarter /
  filltype=alternate
  fillattrs=(color=cxd7d7d7)
  altfillattrs=(color=cxf7f7f7)
  display=(fill values) valuehalign=center;
endinnermargin;
linechart category=month response=actual /
  smoothconnect=true;
endlayout;
end;
run;

/* Generate the graph */
proc sgrender data=sashelp.prdsale template=innermargin;
  format quarter quartername.;
  where year=1994;
```
dynamic year=1994;
run;
PART 4

Plot Statements

Chapter 5
   Key Concepts for Using Plots ........................................... 189

Chapter 6
   Plot Statements ................................................................. 203
Minimum Requirements to Generate a Plot

ODS graphics are generated by template definitions that determine a graph’s layout and appearance and specify the variable roles to be represented in the graph display. A graph can be rendered from a compiled template by associating the template with a data source at run time.

The following SAS program shows the basic structure needed to meet the minimum requirements for generating a plot using GTL:

```sas
proc template;
  define statgraph minimumreq;
    begingraph;
      layout overlay;
        scatterplot x=weight y=height;
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=minimumreq;
run;
```
The DEFINE STATGRAPH statement on PROC TEMPLATE is required to open a definition block for defining and naming a graphics template. The END statement closes the template definition.

A BEGINGRAPH statement block is required to define the outermost container for the graph. The ENDDGRAPH statement closes the block.

At least one layout statement block is required for specifying the elements that compose the graph. To generate a plot, the layout block must contain at least one plot statement. The ENDLAYOUT statement closes the layout block.

The PROC TEMPLATE statement must be run to compile the template and save it in the template store (Sasuser.Templat by default).

The PROC SGRENDER statement is required to produce a graph from a compiled template. The DATA= option specifies a run-time data source to use, and the TEMPLATE= option specifies the template to use. The input data source must satisfy any restrictions that are imposed by the template. For example, it must contain any variables that have been specified on the template’s GTL statements.

**ODS Graphics Environment**

The ODS GRAPHICS statement manages the settings of the ODS Graphics environment and is a statement that you will probably use frequently in your SAS sessions. For example, the ODS GRAPHICS statement provides options that control the physical aspects of your graphs, such as the image size and the name of the image file that is created for the graph.

The default image size of 640 pixels by 480 pixels (4:3 aspect ratio) for ODS Graphics is set in the SAS Registry. You can change the image size using the WIDTH= option, or the HEIGHT= option, or both in the ODS GRAPHICS statement. To name the output image file, use the IMAGENAME= option.

The following ODS GRAPHICS statement sets a 320-pixel width for the graph and names the output image file `modelfit`:

```plaintext
ods graphics / width=320px imagename="modelfit" reset;
proc sgrender data=sashelp.class template=modelfit;
run;
ods graphics off;
```

- The WIDTH= option sets the image width to 320 pixels. Because no HEIGHT= option is used, SAS uses the design aspect ratio of the graph to compute the appropriate height. (The width of 320px is half the default width, so SAS sets the height to 240px, which is half the default height.)
- The IMAGENAME= option sets the name of the output image file to `modelfit`. The RESET option ensures that each time the graph is produced, the previous version of the image file is replaced. Otherwise, image names are incremented (modelfit1, modelfit2, and so on) every time the graph is produced.
In general, it is good practice to specify only one sizing option without the other—just the WIDTH= option or just the HEIGHT= option. That way SAS can maintain the design aspect ratio of the graph, which might be important for many graphs. For example, a graph that has multiple columns or a statistics table on the side needs a wide aspect ratio. Specifying both width and height in such cases might produce unpredictable results.

Note: Size settings in the ODS GRAPHICS statement affect all of the graphs that are rendered in the SAS session, unless they are changed by another ODS GRAPHICS statement. The size for a graph produced by an individual template can be set with the DESIGNWIDTH= and DESIGNHEIGHT= options in the BEGINGRAPH statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement and remain in effect unless they are changed in another ODS GRAPHICS statement or ODS GRAPHICS are turned off.

For more information about using the ODS GRAPHICS statement in GTL, see SAS Graph Template Language: User’s Guide. For a more complete discussion of the ODS GRAPHICS statement, see “ODS GRAPHICS Statement” on page 1483.

Display Attributes

Overview

The display attributes for the lines, colors, marker symbols, and text used in a graph are derived from the ODS style that is in effect when the graph is produced. These display attributes might also be influenced by grouped data. To override default display attributes, all GTL plot statements provide options that manage the graph’s visual appearance. For example, a BOXPLOT statement provides an OUTLIERATTRS= option that manages the visual appearance of outliers.

Two ways are generally available for modifying a graph’s display attributes:

- Change the ODS style that is in effect for the graph. “ODS Styles” on page 17 provides an overview of the use of styles in a graph. SAS Graph Template Language: User’s Guide discusses the use of styles in more detail.
- Override default style settings using GTL statement options. Some examples are given in the sections that follow.

Display Attributes for Non-Grouped Data

Appendix 4, “Display Attributes,” on page 1519 documents the attribute settings that can be specified for the lines, data markers, text, or area fills in a plot. The defaults for these attributes are defined on style elements, but you can use attribute options on the plot statement to change the defaults.

For example, the LINEPARAM statement provides a LINEATTRS= option that specifies the color, line pattern, or line thickness of the plot line. For non-grouped
data, if you do not set a line pattern in your template, then the default line pattern for the plot is obtained from the GraphDataDefault:LineStyle style reference.

To change the default line pattern, a PATTERN= suboption on LINEATTRS= is available. The following table shows the most common line patterns available for the PATTERN= suboption.

**Table 5.1 Commonly Used Line Patterns**

<table>
<thead>
<tr>
<th>Pattern Number</th>
<th>Pattern Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Solid</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ShortDash</td>
<td>················································</td>
</tr>
<tr>
<td>4</td>
<td>MediumDash</td>
<td>···············································</td>
</tr>
<tr>
<td>5</td>
<td>LongDash</td>
<td>···············································</td>
</tr>
<tr>
<td>8</td>
<td>MediumDashShortDash</td>
<td>···············································</td>
</tr>
<tr>
<td>14</td>
<td>DashDashDot</td>
<td>···············································</td>
</tr>
<tr>
<td>15</td>
<td>DashDotDot</td>
<td>···············································</td>
</tr>
<tr>
<td>20</td>
<td>Dash</td>
<td>···············································</td>
</tr>
<tr>
<td>26</td>
<td>LongDashShortDash</td>
<td>···············································</td>
</tr>
<tr>
<td>34</td>
<td>Dot</td>
<td>···············································</td>
</tr>
<tr>
<td>35</td>
<td>ThinDot</td>
<td>···············································</td>
</tr>
<tr>
<td>41</td>
<td>ShortDashDot</td>
<td>···············································</td>
</tr>
<tr>
<td>42</td>
<td>MediumDashDotDot</td>
<td>···············································</td>
</tr>
</tbody>
</table>

"Available Line Patterns" on page 1526 provides the complete list of line patterns that can be used with GTL.

In the following template definition, the LINEPARM statement’s LINEATTRS= option overrides the default line pattern by specifying PATTERN=DASH:

```plaintext
proc template;
define statgraph patternchange;
begingraph;
layout overlay;
  scatterplot y=height x=weight;
  lineparm x=0 y=intercept  slope=slope /
    lineattrs=(pattern=dash);
endlayout;
endgraph;
end;
```
run;

Other display options can be managed the same way. For example, the SCATTERPLOT statement provides a MARKERATTRS= option that specifies the color, size, symbol, and weight of the plot data markers. For non-grouped data, if you do not set a marker symbol in your template, then the default marker symbol is obtained from the GraphDataDefault:MarkerSymbol style reference.

To change the default marker symbol, a SYMBOL= suboption on MARKERATTRS= is available. The following table shows the marker symbols available for the SYMBOL= suboption.

<table>
<thead>
<tr>
<th>Symbol Name</th>
<th>Plot Symbol</th>
<th>Symbol Name</th>
<th>Plot Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrowDown</td>
<td>▼</td>
<td>StarFilled</td>
<td>⭐</td>
</tr>
<tr>
<td>Asterisk</td>
<td>★</td>
<td>Tack</td>
<td>T</td>
</tr>
<tr>
<td>Circle</td>
<td>○</td>
<td>Tilde</td>
<td>∩</td>
</tr>
<tr>
<td>CircleFilled</td>
<td>●</td>
<td>Triangle</td>
<td>△</td>
</tr>
<tr>
<td>Diamond</td>
<td>◊</td>
<td>TriangleFilled</td>
<td>▲</td>
</tr>
<tr>
<td>DiamondFilled</td>
<td>◆</td>
<td>TriangleDown</td>
<td>▼</td>
</tr>
<tr>
<td>GreaterThan</td>
<td>&gt;</td>
<td>TriangleDownFilled</td>
<td>▼</td>
</tr>
<tr>
<td>Hash</td>
<td>#</td>
<td>TriangleLeft</td>
<td>□</td>
</tr>
<tr>
<td>HomeDown</td>
<td>▽</td>
<td>TriangleLeftFilled</td>
<td>▼</td>
</tr>
<tr>
<td>HomeDownFilled</td>
<td>▼</td>
<td>TriangleRight</td>
<td>△</td>
</tr>
<tr>
<td>IBeam</td>
<td>I</td>
<td>TriangleRightFilled</td>
<td>▼</td>
</tr>
<tr>
<td>LessThan</td>
<td>&lt;</td>
<td>Union</td>
<td>U</td>
</tr>
<tr>
<td>Plus</td>
<td>+</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Square</td>
<td>□</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SquareFilled</td>
<td>▼</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>Star</td>
<td>⋆</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In the following template definition, the SCATTERPLOT statement’s MARKERATTRS= option overrides the default marker symbol by specifying SYMBOL=CIRCLEFILLED, which uses a filled circle to represent the data points.

```
proc template;
  define statgraph symbolchange;
    begingraph;
      layout overlay;
        scatterplot y=height x=weight / markerattrs=(symbol=circlefilled);
      endlayout;
    endgraph;
  end;
run;
```

Display Attributes for Grouped Data

Appendix 4, “Display Attributes,” on page 1519 documents the attribute settings that you can specify for the lines, data markers, text, or area fills in a plot. For a grouped plot (that is, when you use the GROUP= option in the plot statement), each distinct group value can be represented in the graph by a different combination of line pattern, color, and marker symbol (depending on the graph type). The defaults for these features are set by the LineStyle, Color, ContrastColor, FillPattern, and MarkerSymbol attributes of the GraphData1–GraphDataN style elements. The MarkerSize and LineThickness style attributes are not honored in the case of grouped data. Table 5.1 on page 192 shows the common line patterns that are available, and Table 5.2 on page 193 shows the marker symbols that are available.

When missing group values are displayed, the default attributes of the missing value are set by the GraphMissing style element unless the MISSING= system option specifies a character other than "." or ",". In that case, missing group value attributes are determined by the GraphData1–GraphDataN style elements.

For grouped plots, attributes such as colors, line patterns, and marker symbols are used to distinguish the individual group values. The attributes are derived from the GraphData1–GraphDataN style elements in the current style. Each group value is assigned attributes from a GraphData1-GraphDataN style element sequentially (1 to N). When plotting a SAS data set, the group-value attributes are assigned in data order. When plotting a CAS in-memory table, the group-value attributes are assigned in ascending order of the group column character or unformatted numeric values. For information about attribute rotation, see "Attribute Rotation Patterns" in SAS Graph Template Language: User’s Guide.

As discussed in “Rotating Visual Attributes for Each Plot in an Overlay” on page 197, plot options might also influence the attribute rotation pattern. For example, in the following template definition, the LINEPARM statement’s LINEATTRS= option specifies PATTERN=DASH. This explicit setting overrides the default line pattern for the plot lines and uses dashed lines for all of the plots, leaving color to distinguish among group values.

```
proc template;
  define statgraph dashedline;
    begingraph;
      layout overlay;
        scatterplot y=height x=weight / group=gender;
      endlayout;
    endgraph;
  end;
run;
```
Rather than setting the same line pattern on all group values, you can change the
default sequence of line patterns that is used for grouped values. To do so, set the
LineStyle attribute in some of the style elements GraphData1–GraphDataN.

In the following example, a style is defined to change the line pattern for
GraphData1 and GraphData2. In this example, the style is derived from the
DEFAULT style. Values are set for the LineStyle attributes in the GraphData1 and
GraphData2 style elements. The first default line in the sequence has long dashes
(style value 6) and the second line has short dashes (style value 4). The LineStyle
settings for the remaining GraphData elements are not set, so are derived from the
parent style (DEFAULT). This new line sequence is used as the default line
sequence for any plot that uses the MyDefault style. To apply the style to a graph,
the STYLE= option is used in the ODS HTML statement to specify the style name.

Here is the code for this example.

/* Specify a path for the ODS output */
filename odsout "output-path";

/* Sort the SASHELP.CLASS data by sex and age. */
proc sort data=sashelp.class(keep=height weight sex age)
   out=class;
   by sex age;
run;

/* Generate slope and intercept data for plot reference lines. */
proc robustreg data=class method=m
   plots=none
   outest=stats(rename=(weight=slope));
   by sex;
   model height=weight;
run;

data class;
   merge class stats(keep=intercept slope sex);
run;

proc template;
   /* Create custom style MYDEFAULT from the MYDEFAULT style. */
   define style MyDefault;
      parent=Styles.Default;
      style GraphData1 from GraphData1 /
        LineStyle=6;
      style GraphData2 from GraphData2 /
        LineStyle=4;
   end;

   /* Create the plot template. */
   define statgraph testPattern;
      begingraph;
      layout overlay;

Similarly, for grouped data, you can set the MarkerSymbol attribute in each of the style elements GraphData1–GraphDataN. In the following example, a style is defined to change the MarkerSymbol attribute for GraphData1–GraphData3. This new sequence is used as the default marker symbol sequence for any grouped plot that uses the MyDefault style.

Here is the code for this example.

```
/* Specify a path for the ODS output */
filename odsout "output-path";

proc template;
  /* Create custom style MyDefault from STYLES.DEFAULT. */
  define style MyDefault;
    parent=Styles.Default;
    MarkerSymbol sequence=("+", "x", "o");
  end;
run;
```

Here is the output.
Overlay-type layouts provide the CYCLEATTRS= option, which specifies whether the default visual attributes of lines, marker symbols, and area fills in nested plot statements automatically change from plot to plot. When CYCLEATTRS=TRUE, all applicable plot statements (SCATTERPLOT, SERIESPLOT, and others) are sequentially assigned the next unused GraphDataN style element. (The sequence is overridden for plot statements that have an explicit setting, either through a style.
element assignment or option settings.) No plot retains its default (implicit) style element.

In the following example, assuming ungrouped data and the default attribute rotation pattern, the series plots are assigned line properties based on the GraphData1, GraphData2, and GraphData3 style elements. The reference line uses GraphReference, not GraphData4.

```
layout overlay / cycleattrs=true;
   seriesplot x=date y=var1;
   seriesplot x=date y=var2;
   seriesplot x=date y=var3;
   referenceline x=cutoff / lineattrs=GraphReference;
endlayout;
```

If one of the plots in this example uses grouped data, then the grouped plots also participate in the default cycles. For example, if the second plot has three groups, then it generates three plots, which are assigned line properties based on the GraphData2, GraphData3, and GraphData4 style elements.

If the plot statement that uses grouped data also uses the INDEX= option to manage the group values (see “Remapping Groups for Grouped Data” on page 198), then the INDEX= option overrides the default behavior. In that case, the grouped plots do not participate in the default cycling.

When one or more of the plots within the layout override the default cycling behavior, the arrangement of the plots within the layout might affect the default mapping of the GraphDataN elements to those statements that participate in the default cycling.

---

**Remapping Groups for Grouped Data**

Indexing can be used to collapse the number of groups that are represented in a graph. For example, if 10 groups are in the data, then indexes 1 and 2 can be assigned to the first two groups, and index 3 can be assigned to all other groups. The third through tenth data groups are treated as a single group in the graph.

Indexing can control the order in which colors, area fills, marker symbols, and line styles are mapped to group values in a graph. This ordering method is needed only for coordinating the data display of multiple graphs when the default mapping would cause group values to be mismatched between graphs.

For example, consider two studies of three drugs, A, B, and C. If Study 1 uses all three drugs, then the first combination of color and marker symbol is mapped to Drug A. The second combination of color and marker symbol is mapped to Drug B, and the third is mapped to Drug C. If Study 2 omits Drug A, then the first combination of color and marker symbol is mapped to Drug B, and the second is mapped to Drug C. If the two graphs are viewed together, then this default mapping causes the group values to be mismatched. The visual attributes that represent Drug A in the first graph represent Drug B in the second graph. Those that represent Drug B in the first graph represent Drug C in the second graph.

The GROUP= option mappings can be made consistent between the two graphs by creating an index column for each study. For these example studies, the GROUP and INDEX columns are the following:
Table 5.3 Study 1

<table>
<thead>
<tr>
<th>Drug1</th>
<th>Index1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 5.4 Study 2

<table>
<thead>
<tr>
<th>Drug2</th>
<th>Index2</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

If the graph for Study 1 specifies INDEX=INDEX1 and the graph for Study 2 specifies INDEX=INDEX2, then the second combination of color and marker symbol is mapped to Drug B in both graphs. The third combination of color and marker symbol is mapped to Drug C in both graphs.

Interactions between Options

When you use GTL statement options to manage the graph display, interactions between options might cause some option settings to be ignored. For example, an ENTRYTITLE statement provides BORDER= and BORDERATTRS= options for managing a border line around the graph title. Border attributes that are set on the BORDERATTRS= option have no effect on the graph title unless the title border line is displayed by setting BORDER=TRUE.

Similarly, if a BOXPLOT statement’s DISPLAY= option suppresses the display of outliers in a box plot, then using the OUTLIERATTRS= option to set outlier attributes has no effect. The OUTLIERATTRS= settings take effect only if DISPLAY= enables the display of outliers.

The option interactions are not limited to options that simply manage visual elements. For example, on a BOXPLOT, if the EXTREME= option extends the box whiskers beyond the fences, then outliers are suppressed in the plot and options that affect the outliers are ignored, if set.

The documentation for each GTL statement identifies the option interactions that might occur on that statement.
Location and Position of Curve Labels

Overview

On plots that generate a curve line (a series plot or a density plot, for example), you can specify a label for the curve line. You can also determine the label's location in the graph. For example, the SERIESPLOT statement provides the following options for managing a curve label:

- CURVELABEL
  specifies a label for the curve line.

- CURVELABELLOCATION
  specifies the location of the curve line label relative to the plot area.

- CURVELABELPOSITION
  specifies the position of the label relative to the curve line.

Curve Label Location Relative to the Plot Area

By default, the label for a curve line is displayed inside the plot area. The following figure shows the default location of the label for a series plot labeled “Curve Label”:

![Curve Label Location=INSIDE](image_url)

Depending on the shape of the curve line, its distribution of values, and the other plot elements that must be displayed within the plot area, GTL might have to add an offset (see “Adjusting Axis Offsets” on page 985) to one of the plot’s axis lines to provide enough room for the curve label. To prevent the offset of the axis line, you can move the curve label outside of the plot area by specifying CURVELABELLOCATION=OUTSIDE on the plot statement:
Regardless of whether the curve label is displayed inside or outside of the plot area, you can use the CURVELABELPOSITION= option to adjust the label’s position relative to the curve line.

**Curve Label Position Relative to the Curve Line**

Given a curve label’s location inside or outside of the plot area, a plot statement’s CURVELABELPOSITION= option can adjust the label’s position relative to the curve line. For example, the following positions are available for a series plot (for some plots, START and END are not available):

- **AUTO** positions the curve label automatically near the end series line along unused axes whenever possible (typically Y2 or X2) to avoid collision with tick values. This position is used only when CURVELABELLOCATION=OUTSIDE.

- **MAX** forces the curve label to appear near maximum series values (typically, to the right).

- **MIN** forces the curve label to appear near minimum series values (typically, to the left).

- **START** forces the curve label to appear near the beginning of the curve. This position is particularly useful when the curve line has a spiral shape. It is used only when CURVELABELLOCATION=INSIDE.

- **END** forces the curve label to appear near the end of the curve. This position is particularly useful when the curve line has a spiral shape. It is used only when CURVELABELLOCATION=INSIDE.

When CURVELABELLOCATION=INSIDE, you can choose whether to position the curve label near the START or END of the curve, or near the minimum data values (MIN) or maximum data values (MAX). START and END use a different algorithm than MIN and MAX. They are particularly useful for spiral-shaped curves whose end points do not correlate with the minimum and maximum data values. In those cases, START or END provides “better” label locations than MIN and MAX.
When `CURVELABELLOCATION=OUTSIDE` and `CURVELABELPOSITION=AUTO`, a "good" position is automatically chosen to avoid collision with the axis information.

The following figure shows the different combinations of label locations and positions:

- The minimum or maximum axis tick marks can be adjusted (see "Adjusting Axis Offsets" on page 985) so that the label can be placed inside the plot area. Increasing label length decreases the area available for displaying plots.
- When `CURVELABELLOCATION=OUTSIDE`, you can set the `CURVELABELPOSITION` to MIN or MAX, but the label might collide with the axis ticks and tick values, unless you are aware of where the axes are positioned.
## Plot Statements

### Dictionary

<table>
<thead>
<tr>
<th>Statement</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXISTABLE Statement</td>
<td>204</td>
</tr>
<tr>
<td>BANDPLOT Statement</td>
<td>204</td>
</tr>
<tr>
<td>BARCHART Statement</td>
<td>221</td>
</tr>
<tr>
<td>BARCHARTPARM Statement</td>
<td>239</td>
</tr>
<tr>
<td>BIHISTOGRAM3DPARM Statement</td>
<td>280</td>
</tr>
<tr>
<td>BLOCKPLOT Statement</td>
<td>319</td>
</tr>
<tr>
<td>BOXPLOT Statement</td>
<td>324</td>
</tr>
<tr>
<td>BOXPLOTPARM Statement</td>
<td>341</td>
</tr>
<tr>
<td>BUBBLEPLOT Statement</td>
<td>377</td>
</tr>
<tr>
<td>CONTOURPLOTPARM Statement</td>
<td>412</td>
</tr>
<tr>
<td>DENDROGRAM Statement</td>
<td>434</td>
</tr>
<tr>
<td>DENSITYPLOTPARM Statement</td>
<td>442</td>
</tr>
<tr>
<td>DROPLINE Statement</td>
<td>450</td>
</tr>
<tr>
<td>ELLIPSE Statement</td>
<td>465</td>
</tr>
<tr>
<td>ELLIPSEPARM Statement</td>
<td>472</td>
</tr>
<tr>
<td>FRINGE PLOT Statement</td>
<td>482</td>
</tr>
<tr>
<td>FRINGEPLOT Statement</td>
<td>492</td>
</tr>
<tr>
<td>HEATMAP Statement</td>
<td>499</td>
</tr>
<tr>
<td>HEATMAPPARM Statement</td>
<td>512</td>
</tr>
<tr>
<td>HISTOGRAM Statement</td>
<td>525</td>
</tr>
<tr>
<td>HISTOGRAM3DPARM Statement</td>
<td>550</td>
</tr>
<tr>
<td>LINECHART Statement</td>
<td>566</td>
</tr>
<tr>
<td>LINEPLOT Statement</td>
<td>583</td>
</tr>
<tr>
<td>LINEPARM Statement</td>
<td>605</td>
</tr>
<tr>
<td>LOESSPLOT Statement</td>
<td>617</td>
</tr>
<tr>
<td>MODELBAND Statement</td>
<td>629</td>
</tr>
<tr>
<td>MOSAICPLOTPARM Statement</td>
<td>640</td>
</tr>
<tr>
<td>NEEDLEPLOT Statement</td>
<td>654</td>
</tr>
<tr>
<td>PBSPLINEPLOT Statement</td>
<td>671</td>
</tr>
<tr>
<td>PIECHART Statement</td>
<td>684</td>
</tr>
<tr>
<td>POLYGONPLOT Statement</td>
<td>699</td>
</tr>
<tr>
<td>REFERENCELINE Statement</td>
<td>728</td>
</tr>
<tr>
<td>REGRESSIONPLOT Statement</td>
<td>742</td>
</tr>
<tr>
<td>SCATTERPLOT Statement</td>
<td>754</td>
</tr>
<tr>
<td>SCATTERPLOTMATRIX Statement</td>
<td>795</td>
</tr>
<tr>
<td>SERIESPLOT Statement</td>
<td>822</td>
</tr>
<tr>
<td>STEEPLOT Statement</td>
<td>865</td>
</tr>
<tr>
<td>SURFACEPLOTPARM Statement</td>
<td>897</td>
</tr>
<tr>
<td>TEXTPLOT Statement</td>
<td>905</td>
</tr>
<tr>
<td>VECTORPLOT Statement</td>
<td>933</td>
</tr>
<tr>
<td>WATERFALLCHART Statement</td>
<td>951</td>
</tr>
</tbody>
</table>
AXISTABLE Statement

Creates an event plot of input data along an axis of an X-Y plot.

Syntax

\texttt{AXISTABLE X=column | expression VALUE=column \langle options\rangle;}
\texttt{AXISTABLE Y=column | expression VALUE=column \langle options\rangle;}

Summary of Optional Arguments

Appearance options

\texttt{CLASS=column | expression}
creates a separate row or column for each unique class value.

\texttt{CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING}
specifies the order in which the class values are displayed.

\texttt{CLUSTERWIDTH=number}
specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

\texttt{COLORGROUP=column | expression | discrete-attr-var}
specifies a column that is used to discretely map the color of the value text.

\texttt{DATATRANSPARENCY=number}
specifies the degree of the transparency of the header, label, and values.

\texttt{DISPLAY=STANDARD | ALL \langle display-options\rangle}
specifies which features to display.

\texttt{DROPONMISSING=TRUE | FALSE}
specifies whether the entire axis table is dropped when all of the VALUE= column values are missing.

\texttt{GUTTER=dimension}
specifies the gap between rows when a class variable is used.

\texttt{INCLUDEMISSINGCLASS=TRUE | FALSE}
specifies whether missing class values are represented in the table.

\texttt{INDENT=dimension}
specifies a value to be used with the INDENTWEIGHT= option to determine the indentation for each text value.

\texttt{INDENTWEIGHT=numeric-column | expression}
specifies the indentation weight (multiplier) for each observation.

\texttt{PAD=dimension \langle pad-options\rangle}
specifies the amount of extra space that is added inside the table border.

\texttt{POSITION=number}
positions the plot along the axis orthogonal to the axis used for the values.

SHOWMISSING=TRUE | FALSE  
specifies whether missing values are represented in the table.

TEXTGROUP=discrete-attr-var  
specifies the discrete attribute variable for a discrete attribute map that  
maps text attributes to values for each observation.

VALUEATTRS=style-element | style-element(text-options) | (text-options)  
specifies the color and font attributes of the text values.

VALUEFORMAT=format  
specifies a SAS format or a user-defined format for the table values.

VALUEHALIGN=AUTO | LEFT | CENTER | RIGHT  
in a Y-axis table, specifies the horizontal alignment of the column values  
relative to the axis-table width.

VALUEJUSTIFY=AUTO | LEFT | CENTER | RIGHT  
specifies the justification of the values in the axis table relative to the axis-  
table width.

Axes options

XAXIS=X | X2  
specifies whether data are mapped to the primary X (bottom) axis or to  
the secondary X2 (top) axis.

YAXIS=Y | Y2  
specifies whether data are mapped to the primary Y (left) axis or to the  
secondary Y2 (right) axis.

Header options

HEADERLABEL="string"  
specifies the text for the table header.

HEADERLABELATTRS=style-element | style-element(text-options) | (text-options)  
specifies the color and font attributes of the table header.

Label options

LABEL="string"  
specifies the text for the table label.

LABELATTRS=style-element | style-element(text-options) | (text-options)  
specifies the color and font attributes of the column label.

LABELHALIGN=AUTO | LEFT | CENTER | RIGHT  
specifies the horizontal alignment of the column label when it is displayed.

LABELJUSTIFY=LEFT | CENTER | RIGHT  
specifies the justification of the column label when it is displayed.

LABELPOSITION=MIN | MAX  
specifies the end of the axis on which the label is displayed.

Midpoint options

CLASSDISPLAY=STACK | CLUSTER  
specifies how the class values are displayed.

Plot reference options

NAME="string"  
assigns a name to this plot statement for reference in other template  
statements.
Statistics options

STAT=AUTO | SUM | MEAN
specifies the statistic that is to be computed for the VALUE= column when
the column is numeric.

Title options

TITLE="string"
specifies the text for the table title.
TITLEATTRS=style-element | style-element(style-options) | (text-options)
specifies the color and font attributes of the table title.
TITLEHALIGN=AUTO | CENTER | LEFT | RIGHT
specifies the horizontal alignment of the title bounding box of the axis
table, relative to the axis table width.
TITLEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the lines of a multiline title in the title bounding
box. The justification is relative to the axis table width.

Required Arguments

Either the X= or the Y= argument must be specified in the AXISTABLE statement.
Specifying X= places an axis table along the X axis of a plot. Specifying Y= places an axis
table along the Y axis of a plot.

X=column | expression
specifies the column for the X axis.

Requirement If not specified, then Y= must be specified.

Y=column | expression
specifies the column for the Y axis.

Requirement If not specified, then X= must be specified.

VALUE=column
specifies the column that contains the axis table values.

Optional Arguments

CLASS=column | expression
creates a separate row or column for each unique class value. Each row or
column is labeled by the class value.

Interaction The DISPLAY= option that is in effect must include LABEL for any
labels to appear.

Note If you specify a column in a SAS data set, the visual attributes for
each class value are assigned in data order. If you specify a column
in a CAS in-memory table, the visual attributes for each class value
are assigned in ascending order of the class column character
values or of unformatted numeric values.

Tip Use the LABELATTRS= option to modify the label text attributes.

CLASSDISPLAY=STACK | CLUSTER
specifies how the class values are displayed.
Note: This feature applied to SAS 9.4M2 and to later releases.

STACK  
displays the class values vertically at each midpoint value on the X axis or horizontally on the Y axis.

CLUSTER  
displays the class values horizontally at each midpoint value on the X axis or vertically on the Y axis.

Restriction  CLUSTER applies only when the axis table is on a discrete axis.

Tip  The CLUSTERWIDTH= option controls the cluster width.

Default  STACK

Interaction  The CLASS= option must be specified for this option to have any effect.

Note  When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

CLASSORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING  
specifies the order in which the class values are displayed.

Note: This feature applies to SAS 9.4M2 and to later releases.

DATA  
displays the class values in the order in which they occur in the data.

REVERSEDATA  
displays the class values in the reverse order from which they occur in the data.

Tip  This option is useful when the plot axis is reversed.

ASCENDING  DESCENDING  
displays the class values in ascending or descending order.

Defaults  DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric class values or on formatted values for character class values.

Interaction  The CLASS= option must be specified for this option to have any effect.

Note  In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory
table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

**CLUSTERWIDTH=** *number*

specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**Range**

0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

**Requirement**

For this option to take effect, the CLASS= option must also be specified, and the CLASSDISPLAY= option must be set to CLUSTER.

**COLORGROUP=** *column | expression | discrete-attr-var*

specifies a column that is used to discretely map the color of the value text.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction**

A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Each unique value of this column is mapped to the COLOR attribute of the GraphData1–GraphDataN style elements that are in effect. If a discrete attribute variable is specified, the color mapping from its associated DISCRETEATTRMAP statement is used.

**Interaction**

This option is ignored when the TEXTGROUP= option is specified.

**Note**

If you specify a column in a SAS data set, colors are mapped to group values in data order. If you specify a column in a CAS in-memory table, colors are mapped to group values in ascending order of the group column character values or unformatted numeric values.

**See**

“DISCRETEATTRVAR Statement” on page 1435

“DISCRETEATTRMAP Statement” on page 1425

**DATATRANSPARENCY=** *number*

specifies the degree of the transparency of the header, label, and values.

**Default**

0

**Range**

0–1, where 0 is opaque and 1 is entirely transparent

**DISPLAY=** **STANDARD | ALL | (display-options)**

specifies which features to display.

**STANDARD**

displays the table values and, if provided, the table label.
ALL
  displays the same features as STANDARD.

(display-options)
  specifies a space-separated list of display options, enclosed in parentheses.
  The following options are supported:

LABEL
  displays the table label.
  The label can be the VALUE= column label or name, the LABEL= value, or the CLASS= value for the table, depending on the options that you specify.

VALUES
  displays the column values.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip: The column values are always displayed, even if DISPLAY=(LABEL) is specified. To hide the table label, specify DISPLAY=(VALUES).

Default: STANDARD

Note: If a table title is specified, it is always displayed.

DROPONMISSING=TRUE | FALSE
  specifies whether the entire axis table is dropped when all of the VALUE= column values are missing.

Note: This feature applies to SAS 9.4M2 and to later releases.

Default: FALSE

Tip: The SHOWMISSING= option controls whether missing values are shown in the table.

See: VALUE= on page 206

“boolean ” on page 1479 for other Boolean values that you can use.

GUTTER=dimension
  specifies the gap between rows when a class variable is used.

Note: This feature applies to SAS 9.4M2 and to later releases.

Defaults: Y-axis table: 8 px
          X-axis table: 0 px

Interaction: The CLASS= option must be specified for this option to have any effect.

See: “dimension” on page 1480
HEADERLABEL="string"
specifies the text for the table header.

Note: Starting with SAS 9.4M2, the HEADERLABEL= option is deprecated and is replaced with the TITLE= option. The syntax and functionality are the same. The HEADERLABEL= option is still honored, but the TITLE= option is preferred.

Default
No table header is displayed

Tip
Use the HEADERLABELATTRS= option to control the appearance of the table header.

HEADERLABELATTRS=style-element | style-element(text-options) | (text-options)
specifies the color and font attributes of the table header.

Note: Starting with SAS 9.4M2, the HEADERLABELATTRS= option is deprecated and is replaced with the TITLEATTRS= option. The syntax and functionality are the same. The HEADERLABELATTRS= option is still honored, but the TITLEATTRS= option is preferred.

See TITLEATTRS= on page 217

INCLUDEMISSINGCLASS=TRUE | FALSE
specifies whether missing class values are represented in the table.

Note: This feature applies to SAS 9.4M2 and to later releases.

Missing class values are included by default. When the data contains missing class values, the label for those values is either blank for missing character values or a dot for missing numeric values.

The following figure shows block plots for classes Class 1, Class 2, and any missing class values.

The following figure shows an X-axis axis table that displays values for classes Class 1, Class 2, and any missing class values.

```
proc format;
  value $missingClass " " = "(Missing)";
  value missingClass . = "(Missing)";
run;
```
A single space enclosed in quotation marks specifies a missing character value and a dot specifies a missing numeric value. Although it might seem appropriate to use empty quotation marks ("" or "") to specify a missing character value, doing so produces unexpected results. To specify a missing character value, enclose a single space in quotation marks (" "). You can use this format for the class columns in the PROC SGRENDER statement. In that case, if the class columns contain missing values, then the labels specified in the format statement are used for the missing classes.

The following figure shows the previous example when format $missingClass is applied to the class variable.

Note: In SAS 9.4M2 and in earlier releases, ODS Graphics does not support Unicode values in user-defined formats. Starting with SAS 9.4M3, ODS Graphics supports Unicode values in user-defined formats only if they are preceded by the (*ESC*) escape sequence. Example: "(*ESC*){unicode beta}". ODS Graphics does not support an escape character that is defined in an ODS ESCAPECHAR statement in user-defined formats.

Default

TRUE

Interaction

The CLASS= option must be specified for this option to have any effect.

See

“boolean ” on page 1479 for other Boolean values that you can use.

**INDENT=** *dimension*

specifies a value to be used with the INDENTWEIGHT= option to determine the indentation for each text value.

Default 1/8 inch

Interaction The INDENTWEIGHT= option must be specified for this option to have any effect.

See “dimension” on page 1480

**INDENTWEIGHT=** *numeric-column | expression*

specifies the indentation weight (multiplier) for each observation.

Interaction For each observation, the INDENT= option value is multiplied by the value of the column specified by this option to determine the indentation for that observation’s value.

**LABEL=** *string*

specifies the text for the table label.

Note: This feature applies to SAS 9.4M2 and to later releases.
The class values are used when the CLASS= option is set and CLASSDISPLAY=STACK is in effect. Otherwise, the VALUE= column label or name is used.

This option is ignored when the CLASS= option is in effect.

If the length of the label exceeds the available space, the label is split on blank space as needed to make it fit.

CLASS= on page 206

LABELATTRS=style-element | style-element(text-options) | (text-options)
specifies the color and font attributes of the column label.

For non-grouped data, the GraphValueText style element.

For grouped data, the label color changes to match the group color derived from the ContrastColor attribute of the GraphData1–GraphDataN style elements.

Group behavior occurs only when the CLASS= and COLORGROUP= option values are the same.

If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphValueText style element.

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

LABELHALIGN=AUTO | LEFT | CENTER | RIGHT
specifies the horizontal alignment of the column label when it is displayed.

This feature applies to SAS 9.4M2 and to later releases.

The alignment is relative to the axis-table width. The axis table width is set by the system to accommodate the maximum width of the values and, when displayed, the label.

AUTO
uses the effective value of the LABELJUSTIFY= option.

LEFT
CENTER
RIGHT
horizontally justifies the label left, center, or right.

The following figure shows these positions for the axis-table label Size.
**LABELJUSTIFY=LEFT | CENTER | RIGHT**

 specifies the justification of the column label when it is displayed.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

The justification is relative to the axis-table width. The axis table width is set by the system to accommodate the maximum width of the values and, when displayed, the label. The following figure shows the justifications LEFT, CENTER, and RIGHT for axis-table label Nominal Size.

```
<table>
<thead>
<tr>
<th>LEFT</th>
<th>CENTER</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal Size</td>
<td>Nominal Size</td>
<td>Nominal Size</td>
</tr>
<tr>
<td>4</td>
<td>Extra Large</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Large</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Small</td>
<td>1</td>
</tr>
</tbody>
</table>
```

**DEFAULT** AUTO

**Restriction** This option applies only to Y-axis tables.

**Interaction** The **DISPLAY=** option must include **LABEL** for this option to have any effect.

**LABELPOSITION=MIN | MAX**

 specifies the end of the axis on which the label is displayed. The label is aligned with the tick values on the axis. The following figure shows the MIN and MAX positions for Y axis and X-axis tables.

```
<table>
<thead>
<tr>
<th>LEFT</th>
<th>CENTER</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal Size</td>
<td>Nominal Size</td>
<td>Nominal Size</td>
</tr>
<tr>
<td>4</td>
<td>Extra Large</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Large</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Small</td>
<td>1</td>
</tr>
</tbody>
</table>
```

**DEFAULT** CENTER

**Restriction** This option applies only to Y-axis tables.

**Interaction** The **DISPLAY=** option must include **LABEL** for this option to have any effect.
### NAME="string"

Assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**

- The string is case sensitive, cannot contain spaces, and must define a unique name within the template.
- This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

### PAD=dimension | (pad-options)

Specifies the amount of extra space that is added inside the table border.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

- **dimension**
  - Specifies a dimension to use for the extra space at the left, right, top, and bottom of the table border.
- **(pad-options)**
  - Specifies a space-separated list of one or more name-value-pair pad options, enclosed in parentheses.
  - The list can contain one or more of the following pad options:

  **LEFT=dimension**
  - Specifies the amount of extra space added to the left side.
    - **Default:** 4 px
    - **Restriction:** This option applies only to Y-axis tables.

  **RIGHT=dimension**
  - Specifies the amount of extra space added to the right side.
    - **Default:** 4 px
    - **Restriction:** This option applies only to Y-axis tables.
TOP=dimension
specifies the amount of extra space added to the top.

Default 0 px

Restriction This option applies only to X-axis tables.

BOTTOM=dimension
specifies the amount of extra space added to the bottom.

Default 0 px

Restriction This option applies only to X-axis tables.

Note Sides that are not assigned padding are padded with the default amount of space.

Tip Use pad-options to create non-uniform padding.

Note The default units for dimension are pixels.

See "dimension" on page 1480

POSITION=number
positions the plot along the axis orthogonal to the axis used for the values. This option enables you to position the plot when the AXISTABLE statement is not placed in an INNERMARGIN block.

number specifies the position on the orthogonal axis as a fraction of the axis range.

When using the VALUEHALIGN option to align the values, in some cases, the axis-table values might not align properly within the space allocated for the axis table. For example, right-justifying the values might cause the values to overlap other graphical elements. In that case, rather than allocating more space for the axis table, you can use the POSITION= option to move the axis table to the left along the X axis as shown in the following figure.

<table>
<thead>
<tr>
<th>Default</th>
<th>POSITION=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Class</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Default Determined by the system.

Range 0 (bottom)–1 (top)

Interaction This option is ignored when the AXISTABLE statement is placed in an INNERMARGIN block. It is also ignored when the AXISTABLE statement is placed in a LAYOUT OVERLAY block by itself.
Tip
To reserve space for the plot at either end of the orthogonal axis, specify the OFFSETMIN= or OFFSETMAX= option for the orthogonal axis.

SHOWMISSING=TRUE | FALSE
specifies whether missing values are represented in the table.

Note: This feature applies to SAS 9.4M2 and to later releases.

The values are evaluated before the column format is applied. When this option is set to FALSE, missing numeric and character values are hidden.

Default TRUE

See “boolean ” on page 1479 for other Boolean values that you can use.

STAT=AUTO | SUM | MEAN
specifies the statistic that is to be computed for the VALUE= column when the column is numeric.

AUTO
computes the SUM statistic when the VALUE= column is numeric.

When the column is character, it uses the first column value as the statistic value.

SUM
MEAN
computes the SUM or MEAN statistic when the VALUE= column is numeric.

When the column is character, it uses the first column value as the statistic value.

Default AUTO

Interaction When the VALUE= column is character, the STAT= option always uses the first column value as the statistic value. In that case, SUM and MEAN are ignored.

TEXTGROUP=discrete-attr-var
specifies the discrete attribute variable for a discrete attribute map that maps text attributes to values for each observation. The discrete attribute variable is defined in a DISCRETEATTRVAR statement.

Restrictions A discrete attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

The SIZE= specification in the discrete attribute map TEXTATTRS= option is ignored.

Interaction When this option is specified, the COLORGROUP= option is ignored.

TITLE=“string”
specifies the text for the table title.

Note: This feature applies to SAS 9.4M2 and to later releases.
<table>
<thead>
<tr>
<th>Default</th>
<th>Note</th>
<th>Tip</th>
</tr>
</thead>
<tbody>
<tr>
<td>No table title is displayed</td>
<td>When an axis table is specified in the prototype of a data-driven layout, if the table is on the X axis, then the table title appears only in the first column of each row. If the table is on the Y axis, then the table title appears only in the first row of each column.</td>
<td>Use the TITLEATTRS= option to control the appearance of the table title.</td>
</tr>
</tbody>
</table>

**TITLEATTRS=**

`style-element` | `style-element(text-options)` | (text-options)

specifies the color and font attributes of the table title.

- **Note:** This feature applies to SAS 9.4M2 and to later releases.
- **Default**: The GraphLabelText style element.
- **Interaction**: If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphLabelText style element.
- **See**: "General Syntax for Attribute Options" on page 1519 for the syntax for using a `style-element`.
  "Text Options" on page 1525 for available `text-options`.

**TITLEHALIGN=**

`AUTO` | `CENTER` | `LEFT` | `RIGHT`

specifies the horizontal alignment of the title bounding box of the axis table, relative to the axis table width.

- **Note:** This feature applies to SAS 9.4M2 and to later releases.
- **AUTO**: aligns the title bounding box based on the table type:
  - For a Y-axis table, aligns the title bounding box according to the effective `TITLEJUSTIFY=` option value.
  - For an X-axis table, aligns the title bounding box LEFT.
- **CENTER** | **LEFT** | **RIGHT**

  horizontally aligns the table title bounding box center, left, or right.

- **Default**: AUTO
- **Restriction**: In SAS 9.4M2, this option applies only to Y-axis tables. Starting with SAS 9.4M3, this option applies to Y-axis tables and to X-axis tables.
- **Interaction**: The `TITLE=` option must be specified for this option to have any effect.
- **Tip**: For a multiline title in a Y-axis table, use `TITLEJUSTIFY=` to justify the title text lines in the title bounding box.
TITLEJUSTIFY=LEFT | CENTER | RIGHT
specifies the justification of the lines of a multiline title in the title bounding box. The justification is relative to the axis table width.

Note: This feature applies to SAS 9.4M2 and to later releases.

Default LEFT

Restrictions This option applies only to Y-axis tables.

The title must wrap to multiple lines for this option to have any effect.

Interaction The TITLE= option must be specified for this option to have any effect.

Tip Use TITLEHALIGN= to align the title bounding box.

VALUEATTRS=style-element | style-element(text-options) | (text-options)
specifies the color and font attributes of the text values.

Default The GraphDataText style element.

Interaction If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the properties that are not specified are derived from the GraphLabelText style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

VALUEFORMAT=format
specifies a SAS format or a user-defined format for the table values.

Note: This feature applies to SAS 9.4M2 and to later releases.

Default The format that is in effect for the column specified in the VALUE= option. If no format is in effect, BEST6 is used for numeric columns.

Note Not all of the SAS formats are supported. See Appendix 5, “SAS Formats Not Supported,” on page 1541.

VALUEHALIGN=AUTO | LEFT | CENTER | RIGHT
in a Y-axis table, specifies the horizontal alignment of the column values relative to the axis-table width. The axis table width is set by the system to accommodate the maximum width of the values and, when displayed, the label.

Note: This feature applies to SAS 9.4M2 and to later releases.

AUTO
uses the effective value of the VALUEJUSTIFY= option.
LEFT
CENTER
RIGHT

aligns the values left, center, or right relative to the axis-table width.

The following figure shows an example of each of these alignments.

<table>
<thead>
<tr>
<th>LEFT</th>
<th>CENTER</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Class</td>
<td>Class</td>
</tr>
<tr>
<td>4 - D</td>
<td>4 - D</td>
<td>4 - D</td>
</tr>
<tr>
<td>3 - C</td>
<td>3 - C</td>
<td>3 - C</td>
</tr>
<tr>
<td>2 - B</td>
<td>2 - B</td>
<td>2 - B</td>
</tr>
<tr>
<td>1 - A</td>
<td>1 - A</td>
<td>1 - A</td>
</tr>
</tbody>
</table>

Default: AUTO

Restriction: This option applies only to Y-axis tables.

Tip: If the values do not align properly within the space allocated for the axis table, use the POSITION= option to move the axis table along the X axis as needed.

VALUEJUSTIFY=AUTO | LEFT | CENTER | RIGHT

specifies the justification of the values in the axis table relative to the axis-table width. The axis table width is set by the system to accommodate the maximum width of the values and, when displayed, the label.

Note: This feature applies to SAS 9.4M2 and to later releases.

AUTO

uses LEFT for text values or RIGHT for numeric values.

CENTER
LEFT
RIGHT

horizontally aligns the table values center, left, or right, relative to the axis-table width.

The following figure shows an example of each of these alignments.

<table>
<thead>
<tr>
<th>LEFT</th>
<th>CENTER</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Class</td>
<td>Class</td>
</tr>
<tr>
<td>4 - ABCD</td>
<td>4 - ABCD</td>
<td>4 - ABCD</td>
</tr>
<tr>
<td>3 - ABC</td>
<td>3 - ABC</td>
<td>3 - ABC</td>
</tr>
<tr>
<td>2 - AB</td>
<td>2 - AB</td>
<td>2 - AB</td>
</tr>
<tr>
<td>1 - A</td>
<td>1 - A</td>
<td>1 - A</td>
</tr>
</tbody>
</table>

Default: AUTO

Restriction: This option applies only to Y-axis tables.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
Defaults | Interaction
--- | ---
The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

**YAXIS=** Y | Y2
--- | ---
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Defaults | Interaction
--- | ---
Y

Details
The AXISTABLE statement enables you to place text values along the X or Y axis at specific values on the axis. It offers more flexibility than the BLOCKPLOT statement, which is used to denote changes in block values along the axis. The X and Y data does not need to be sorted.

Example: AXISTABLE Statement
This example shows how to add a table of average sales data by division below a bar chart of total sales by product and country. Here is the output that is generated by this example.

An inner margin is created at the bottom of the layout container to reserve space for the table. An AXISTABLE statement is used in the INNERMARGIN block to show...
the average sales by division for each product. Here is the SAS code for this example.

```sas
proc template;
define statgraph axistable;
begingraph;
entrytitle "Average Product Sales By Division and Country";
layout overlay / cycleattrs=true
yaxisopts=(offsetmax=0.15 label="Sales By Country");
innermargin / align=bottom opaque=true backgroundcolor=cxf5f5f5;
axistable x=product value=actual /
   name="division" stat=mean display=(label)
   headerlabel="Sales By Division"
   headerlabelattrs=GraphLabelText
   valueattrs=(size=9pt weight=bold)
   colorgroup=division class=division;
endinnermargin;
barchart category=product y=actual / name="country"
   barlabel=true barlabelformat=dollar5.0
   stat=mean group=country groupdisplay=cluster;
discretelegend "country" / title="Country:" location=inside
   valign=top;
endlayout;
endgraph;
end;
run;
proc sgrender data=sashelp.prdsale template=axistable;
run;
```

See Also

"Creating an Axis-Aligned Inset with an Axis Table" in SAS Graph Template Language: User’s Guide

**BANDPLOT Statement**

Creates a band plot that typically shows confidence or prediction limits.

**Requirements:**

You must specify either an X= argument or a Y= argument. You cannot specify both. When you specify the X argument, you must also specify LIMITLOWER and LIMITUPPER arguments for Y values. When you specify the Y argument, you must also specify LIMITLOWER and LIMITUPPER arguments for X values. The plot data should be sorted by the X or Y variable that is used in the BANDPLOT statement. Otherwise, specify CONNECTORDER= AXIS in the BANDPLOT statement.

**Syntax**

```sas
BANDPLOT X=column | expression
LIMITLOWER=number | numeric-column | expression
LIMITUPPER=number | numeric-column | expression <options>;
```
**BANDPLOT**  
\[ Y = \text{numeric-column} \mid \text{expression} \]  
LIMITLOWER = \text{number} \mid \text{numeric-column} \mid \text{expression}  
LIMITUPPER = \text{number} \mid \text{numeric-column} \mid \text{expression} <\text{option(s)}>;

**Summary of Optional Arguments**

**Appearance options**

- **ANTIALIAS=** AUTO \mid OFF  
  specifies whether anti-aliasing is turned off for this plot.

- **CONNECTORDER=** VALUES \mid AXIS  
  specifies how to connect the data points to form the band lines.

- **CURVELABELATTRS=** style-element \mid style-element (text-options) \mid (text-options)  
  specifies the color and font attributes of the band labels.

- **DATATRANSPARENCY=** number  
  specifies the degree of the transparency of the band fill and band outline.

- **DISPLAY=** (display-options) \mid STANDARD \mid ALL  
  specifies which band features to display.

- **EXTEND=** TRUE \mid FALSE  
  specifies whether the constant or "step" band is to be drawn to the area bounded by the axes.

- **FILLATTRS=** style-element \mid style-element (fill-options) \mid (fill-options)  
  specifies the appearance of the filled band area.

- **FILLPATTERNATTRS=** style-element \mid (fill-pattern-options)  
  specifies the appearance of the pattern-filled areas.

- **INDEX=** positive-integer-column \mid expression  
  specifies indices for mapping band attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

- **JUSTIFY=** LEFT \mid CENTER \mid RIGHT  
  specifies the location of the data point relative to the step when TYPE=STEP.

- **OUTLINEATTRS=** style-element \mid style-element (line-options) \mid (line-options)  
  specifies the appearance of the band outlines.

- **TYPE=** SERIES \mid STEP  
  specifies how the data points for lower and upper band boundaries are interpolated.

**Axes options**

- **XAXIS=** X \mid X2  
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

- **YAXIS=** Y \mid Y2  
  specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Data tip options**

- **ROLENAME=** (role-name-list)  
  specifies user-defined roles that can be used to display information in the data tips.

- **TIP=** (role-list) \mid NONE
specifies the information to display when the cursor is positioned over the band plot.

TIPFORMAT=(role-format-list)
    specifies display formats for tip columns.

TIPLABEL=(role-label-list)
    specifies display labels for tip columns.

Label options

CURVELABELLOCATION=INSIDE | OUTSIDE
    specifies the location of the band labels relative to the plot area.

CURVELABELLOWER="string" | column
    specifies a label for the lower band limit.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
    specifies the position of the band labels relative to the curve line.

CURVELABELSPLIT=TRUE | FALSE
    specifies whether to split the band labels at the specified split characters.

CURVELABELSPLITCHAR="character-list"
    specifies one or more characters on which the band labels can be split if needed.

CURVELABELSPLITCHARDROP=TRUE | FALSE
    specifies whether the split characters are included in the band label text.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
    specifies the justification of the strings that are inside the band label blocks.

CURVELABELUPPER="string" | column
    specifies a label for the upper band limit.

LEGENDLABEL="string"
    specifies a label to be used in a discrete legend for this plot.

Midpoint options

GROUP=column | discrete-attr-var | expression
    creates a separate band plot for each unique group value of the specified column.

INCLUDEMISSINGGROUP=TRUE | FALSE
    specifies whether missing values of the group variable are included in the plot.

Plot reference options

MODELNAME="plot-name"
    specifies the name of the plot from which to derive the interpolation for the band.

NAME="string"
    assigns a name to this plot statement for reference in other template statements.

Required Arguments

You must specify either an X= or Y= argument. You cannot specify both. In addition, the LIMITLOWER= and LIMITUPPER= arguments must be used to specify the lower and upper lines for the band.

X=column | expression
    specifies X values. Numeric or character values can be used.
Requirement: You must also specify the LIMITLOWER= and the LIMITUPPER= arguments for the Y values.

**Y=column | expression**

Specifies Y values. Numeric or character values can be used.

Requirement: You must also specify the LIMITLOWER= and the LIMITUPPER= arguments for the X values.

**LIMITLOWER=number | numeric-column | expression**

Specifies a constant or column representing the values of the lower band line.

Interactions: When this option is used with the X= option, it specifies the Y value or values.

When this option is used with the Y= option, it represents the X value or values.

Note: If a constant is specified, then a straight line is drawn.

**LIMITUPPER=number | numeric-column | expression**

Specifies a constant or column representing the values of the lower band line.

Interactions: When this option is used with the X= option, it specifies the Y value or values.

When this option is used with the Y= option, it represents the X value or values.

Note: If a constant is specified, then a straight line is drawn.

Optional Arguments

**ANTIALIAS=AUTO | OFF**

Specifies whether anti-aliasing is turned off for this plot.

Note: This feature applies to SAS 9.4M2 and to later releases.

**AUTO**

Specifies that anti-aliasing is controlled by the ANTIALIAS= option in the ODS GRAPHICS statement.

**OFF**

Specifies that anti-aliasing is always disabled for this plot.

Default: AUTO

Interaction: This option overrides the ANTIALIAS= option in the ODS GRAPHICS statement.

**CONNECTORDER=VALUES | AXIS**

Specifies how to connect the data points to form the band lines.

**VALUES**

Connects data points in the order read from the X column (or Y column).

Note: In SAS 9.4M6 and earlier releases, unexpected results might occur when VALUES is specified and data stored in a CAS in-memory table.
is used. In that case, use AXIS instead. Starting with SAS Viya 3.5 and SAS Studio 5.2, VALUES is ignored when data stored in a CAS in-memory table is used.

**AXIS**

connects data points as they occur left-to-right along the X axis (or bottom-to-top along the Y axis).

**Tip** You can use this value to ensure the expected connect order for certain types of series lines (for example, time series) when the input data might not be sorted by the X column (or Y column).

**Defaults**

VALUES if the data is a SAS data set

AXIS if the data is a CAS in-memory table

**CURVELABELATR**

**LATRS=style-element | style-element (text-options) | (text-options)**

specifies the color and font attributes of the band labels.

**Defaults**

For non-grouped data, the GraphValueText style element.

For grouped data, text color is derived from the GraphData1:ContrastColor–GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

**Note** When you specify style-element, only the style attributes COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT are used.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**CURVELABELLOW**

**ER=string | column**

specifies a label for the lower band limit.

**Default**

No curve label is displayed for the lower band

**Requirements**

For non-grouped data, use "string".

For grouped data, use a column to define the lower band labels for each group value. All of the labels for a specific group value must be the same. Otherwise, the results are unpredictable.

**Tip**

The font and color attributes for the label are specified by the CURVELABELATR**LRS= option.

**CURVELABELUPPER=**

**string | column**

specifies a label for the upper band limit.

**Default**

No curve label is displayed for the upper band

**Requirements**

For non-grouped data, use "string".
For grouped data, use a column to define the upper band labels for each group value. All of the labels for a specific group value must be the same. Otherwise, the results are unpredictable.

**Tip**
The font and color attributes for the label are specified by the CURVELABELATTRS= option.

**CURVELABELLOCATION=**INSIDE | OUTSIDE
specifies the location of the band labels relative to the plot area.

**INSIDE**
locates the labels inside the plot area

**OUTSIDE**
locates the labels outside the plot area

**Default** INSIDE

**Restriction** OUTSIDE cannot be used when the BANDPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE where axes might be external to the grid.

**Interactions** This option is used with the CURVELABELPOSITION= option to determine where the band labels appear. For more information, see "Location and Position of Curve Labels" on page 200.

The EXTEND= option is ignored when CURVELABELLOCATION=INSIDE is in effect.

**CURVELABELPOSITION=**AUTO | MAX | MIN | START | END
specifies the position of the band labels relative to the curve line.

**AUTO**
positions the band labels automatically near the band boundary along unused axes whenever possible (typically Y2 and X2).

**Restriction** This option is used only when CURVELABELLOCATION=OUTSIDE.

**MAX**
forces the band labels to appear near maximum band values (maximum X-values for horizontal curves, and maximum Y-values for vertical curves).

**MIN**
forces the band label to appear near minimum band values (minimum X-values for horizontal curves, and minimum Y-values for vertical curves)

**START**
forces band labels to appear near the beginning of the curve.

**Restriction** This option is used only when CURVELABELLOCATION=INSIDE.

**Tip** This option is particularly useful when the curve line has a spiral shape.

**END**
forces band labels to appear near the end of the curve.
Restriction  This option is used only when CURVELABELLOCATION=INSIDE.

Tip  This option is particularly useful when the curve line has a spiral shape.

Defaults  AUTO when CURVELABELLOCATION=OUTSIDE.

END when CURVELABELLOCATION=INSIDE.

Restrictions  The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

The BANDPLOT statement does not support the BEGINGRAPH statement option LABELPLACEMENT=SA on page 33.

Interaction  This option is used with the CURVELABELLOCATION= option to determine where the band labels appear. For more information, see “Location and Position of Curve Labels” on page 200.

Note  When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the band label might fall outside of the graph area. In that case, the band label might not be displayed or might be positioned incorrectly.

Tip  When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.

**CURVELABELSPLIT=TRUE | FALSE**

specifies whether to split the band labels at the specified split characters. When a band label is split, the label is split on each occurrence of the specified split characters.

Default  FALSE. The band label is not split.

Requirement  The CURVELABEL= option must also be specified.

Interactions  The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.

This option has no effect when CURVELABELPOSITION=AUTO.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITCHAR=“character-list”**

specifies one or more characters on which the band labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the band label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the band labels are split unconditionally at each occurrence of any of the specified split characters. If the band label does not contain any of the specified characters, then the label is not split.
"character-list"  
one or more characters with no delimiter between each character and 
enclosed in quotation marks.

Default  
A blank space

Requirements  
The list of characters must be specified with no delimiters. For 
example, to specify the split characters a, b, and c, use the 
following option:

```
curvelabelsplitchar="abc"
```

The CURVELABEL= option and the CURVELABELSPLIT=TRUE 
option must also be specified.

Interactions  
This option has no effect if CURVELABELPOSITION=AUTO.

The CURVELABELSPLITCHARDROP= option specifies whether 
the split characters are included in the band labels or are 
dropped.

Notes  
When multiple characters are specified, the order of the 
characters in the list is not significant.

The split characters are case sensitive.

Tip  
Use the CURVELABELSPLITJUSTIFY= option to specify the 
justification of the strings in the band label blocks.

**CURVELABELSPLITCHARDROP=TRUE | FALSE**  
specifies whether the split characters are included in the band label text.

**TRUE**  
drops the split characters from the band label text.

**FALSE**  
includes the split characters in the band label text.

When CURVELABELSPLIT=TRUE and 
CURVELABELSPLITCHARDROP=FALSE, each split character remains as 
the last character in the current line. The characters that follow the split 
character, up to and including the next split character, are then wrapped to 
the next line.

The following figure shows an example of a band label with the following 
specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

**Note:** The horizontal line to the left of the label represents the maximum end of 
the band for reference.
When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default: TRUE. The split characters are dropped from the band labels.

Requirement: The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction: The CURVELABELSPLITCHAR= option specifies the split characters.

See: “boolean” on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITJUSTIFY=** AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the band label blocks.

- **AUTO**
  - justifies the labels based on the CURVELABELPOSITION= option.

<table>
<thead>
<tr>
<th>CURVELABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

- **CENTER**
- **LEFT**
- **RIGHT**
  - justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

**Note:** The horizontal line to the left of each label represents the maximum end of the band for reference.

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

Default: AUTO
**Requirement**
The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction**
This option has no effect if CURVELABELPOSITION=AUTO.

**DATATRANSPARENCY=** *number*

specifies the degree of the transparency of the band fill and band outline.

**Default**
0

**Range**
0–1, where 0 is opaque and 1 is entirely transparent

**Note**
This option does not affect the band curve labels.

**Tip**
The FILLATTRS= option can be used to set transparency for just the band area. You can combine this option with FILLATTRS= to set one transparency for the band outline but a different transparency for the band fill. Example:

data transparency=0.2 fillattrs=(transparency=0.6)

**DISPLAY=** *(display-options)* | STANDARD | ALL

specifies which band features to display.

*(display-options)*
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

FILL
displays the band fill color

FILLPATTERN
displays the band fill pattern

**Note**
This feature applies to **SAS 9.4M5** and to later releases.

**Tip**
For this option to have any effect, the active ODS style must specify a fill pattern or a fill pattern must be specified with the FILLPATTERNATTRS= option.

**OUTLINE**
displays the band outline

**STANDARD**
specifies FILL only

**ALL**
specifies all features: FILL, FILLPATTERN, and OUTLINE

**Default**
The GraphBand:DisplayOpts style reference.

**Tip**
Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the band outline and fill color. Starting with **SAS 9.4M5**, use the FILLPATTERNATTRS= option to control the appearance of the band fill pattern.

**EXTEND=** TRUE | FALSE

specifies whether the constant or "step" band is to be drawn to the area bounded by the axes.
**Default**

FALSE

**Requirement**

When this option is used for a constant band, constants must be specified for the upper and lower band limits. This requirement does not apply to "step" bands.

**Interactions**

This option is ignored when band labels are placed inside the plot area (CURVELABELLOCATION=INSIDE). To extend the bands in that case, use the CURVELABELLOCATION=OUTSIDE option.

If the X or Y value is character, then the EXTEND= option is honored only when the upper and lower limits specify a number.

**Tip**

If this option is not specified, then there can be a small gap between the line and the axis. The gap is controlled by the axis offset. If the axis offset is set to 0, then there is no gap.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

---

**FILLATTRS=**

```plaintext
style-element | style-element (fill-options) | (fill-options)
```

specifies the appearance of the filled band area.

**Defaults**

For non-grouped data, the GraphConfidence:Color style reference.

For grouped data, the Color attribute of GraphData1–GraphDataN style elements.

**Interaction**

For this option to have any effect, the fill must be enabled by the ODS style or by the DISPLAY= option.

**Tip**

The DATATRANSAPRENCY= option sets the transparency for both the band fill and band outline. You can combine this option with DATATRANSAPRENCY= to set one transparency for the band outline and a different transparency for the band fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**See**

"General Syntax for Attribute Options" on page 1519 for the syntax for using a style-element.

"Fill Color Options" on page 1520 for available fill-options values.

---

**FILLPATTERNATTRS=**

```plaintext
style-element | (fill-pattern-options)
```

specifies the appearance of the pattern-filled areas.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Defaults**

For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

**Interaction**

For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.
Tip  In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Pattern Options” on page 1521 for available fill-pattern-options.

GROUP=column | discrete-attr-var | expression
creates a separate band plot for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction  A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

A distinct band is created for each group value by varying the visual attributes of the band display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern 2</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern. If the active ODS style does not specify FillPattern, fill patterns are not displayed by default.</td>
</tr>
<tr>
<td>Outline</td>
<td>ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
</tbody>
</table>

1 The default display features are determined by the DisplayOpts attribute of the GraphBand style element.
2 Fill patterns are valid for band plots starting with SAS 9.4M5.

Restrictions  This option can be used only when a numeric column is specified for the upper limit or the lower limit of the band plot. The other limit could be a constant, if desired.

Starting with SAS 9.4M5, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS
The statement option `LINEPATTERNOBSMAX=`. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase `LINEPATTERNOBSMAX=` to the value suggested in the note or change the line pattern to `SOLID`.

**Interactions**

To label grouped band plots, you must specify `CURVELABELLOWER=` *column* and `CURVELABELUPPER=` *column*

The group values are mapped in the order of the data, unless the `INDEX=` option is used to alter the default sequence of colors and line patterns.

The `INCLUDEMISSINGGROUP` option controls whether missing group values are considered a distinct group value.

**Note**

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tip**

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the band lines, but the `PATTERN=` suboption of the `OUTLINEATTRS=` option could be used to assign the same line pattern to all band outlines.

**See**

“DISCRETEATTRVAR Statement” on page 1435

**INCLUDEDMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the group variable are included in the plot.

**Default**

`TRUE`

**Interaction**

For this option to take effect, the `GROUP=` option must also be specified.

**Tip**

The attributes of the missing group value are determined by the `GraphMissing` style element unless a discrete attribute map is in effect, the `INDEX=` option is used, the `MISSING=` system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a `GraphData1`–`GraphDataN` style element instead of by the `GraphMissing` style element.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**

specifies indices for mapping band attributes (fill and outline) to one of the `GraphData1`–`GraphDataN` style elements.

**Requirements**

The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.
The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Notes**
The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip**
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 198.

**JUSTIFY**=LEFT | CENTER | RIGHT
specifies the location of the data point relative to the step when TYPE=STEP.

| ![Diagram of data points at left, center, and right of steps] |
|-------------|-------------|-------------|
| LEFT | CENTER | RIGHT |

**Default**
LEFT

**Requirement**
TYPE= STEP must also be specified for this option to have any effect.

**Interaction**
If the MODELNAME= option is specified, then this option is ignored.

**LEGENDLABEL**="string"
specifies a label to be used in a discrete legend for this plot.

**Default**
The string specified on the NAME= option.

**Restriction**
This option applies only to an associated DISCRETELEGEND statement.

**Interaction**
If the GROUP= option is specified, then this option is ignored.

**MODELNAME**="plot-name"
specifies the name of the plot from which to derive the interpolation for the band. When this option is used, the band plot forms prediction or confidence limits for the plot that supplies the fitted model.

**Requirement**
plot-name must be the name that has been assigned on the associated plot's NAME= option.
Interaction  This option overrides the JUSTIFY= and TYPE= options.

Tip  If this option is not specified, then the interpolation is set by the TYPE= option.

**NAME="string"**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction  The string is used as the default legend label if the LEGENDLABEL= option is not used.

**OUTLINEATTRS=**

specifies the appearance of the band outlines.

Defaults  For non-grouped data, the GraphConfidence style element.

For grouped data, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements.

Restriction  Starting with SAS 9.4M5, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID.

Interaction  For this option to have any effect, the outline must be enabled by the ODS style or by the DISPLAY= option.

See  “General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element.

“Line Options” on page 1522 for available line-options values.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

( role-name-list )

a space-separated list of role-name = column pairs.

Example  The following example assigns the column Obs to the user-defined role TIP:

ROLENAMEx(TIP1=OBS)

Default  No user-defined roles
The role names that you choose must be unique and different from the predefined roles X, Y, LIMITUPPER, LIMITLOWER, GROUP, CURVELABELUPPER, and CURVELABELLOWER.

TIP=(role-list) | NONE

specifies the information to display when the cursor is positioned over the band plot. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the band plot can be specified along with roles that do.

(role-list)

an ordered, space-separated list of unique BANDPLOT and user-defined roles. BANDPLOT roles include X, Y, LIMITUPPER, LIMITLOWER, GROUP, INDEX, CURVELABELUPPER, and CURVELABELLOWER.

User-defined roles are defined with the ROLENAME= option.

Note

CURVELABELUPPER and CURVELABELLOWER are considered roles only when they are assigned a column of values. They are not considered roles and do not display data tips when assigned a string.

Example

This example displays data tips for the columns assigned to the roles X, LIMITUPPER, and LIMITLOWER as well as the column Obs, which is not assigned to any pre-defined BANDPLOT role. The Obs column must first be assigned a role.

ROLENAME=(TIP1=OBS)
TIP=(TIP1 X LIMITUPPER LIMITLOWER)

NONE

suppresses data tips from the plot.

Default

The columns assigned to these roles are automatically included in the data tip information: X, Y, LIMITUPPER, LIMITLOWER, and GROUP.

Requirement

To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip

The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)

a space-separated list of role-name = format pairs.

Example

ROLENAME=(TIP1= SALARY)
TIP=(TIP1)
**TIFORMAT**=(TIP1=DOLLAR12.)

**Default**
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

---

**TIPLABEL**=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

**role-label-list**
a space-separated list of rolename ="string" pairs.

**Example**
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

**Default**
The column label or column name of the column assigned to the role.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

---

**TYPE=SERIES | STEP**
specifies how the data points for lower and upper band boundaries are interpolated.

**SERIES**
connects the data points using line segments (as in a SeriesPlot).

**STEP**
connects the data points (as in a StepPlot).

**Default**
SERIES

**Interactions**
TYPE=STEP must be specified to enable the JUSTIFY= option.

If the MODELNAME= option is specified, then this option is ignored.

---

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default**
X

**Interactions**
This option is ignored if the X= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.
**YAXIS=** Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Default**
Y

**Interactions**
This option is ignored if the Y= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**Details**
A band plot can specify an X column with Y upper and lower limits, or a Y column with X upper and lower limits. If you specify the X argument, then you must specify LIMITLOWER and LIMITUPPER arguments for the Y values to apply the limits to the Y axis. If you specify the Y argument, then you must specify LIMITLOWER and LIMITUPPER arguments for the X values to apply the limits to the X axis.

When you use a BANDPLOT statement to display prediction or confidence limits, the band plot can be used with another plot that specifies a fitted model. For example, it can be used with a series or step plot. In these cases, use the BANDPLOT option MODELNAME= or TYPE= to identify the interpolation for the band.

You can use the BANDPLOT statement in displays that are independent of other plots. For example, a band plot can be used to define yellow and green areas in an OVERLAY LAYOUT statement that also contains a scatter plot. This use implies concern for any of the scatter plot values that fall in the yellow area and comfort for any values that fall in the green area. For this use, the upper and lower limits would be specified by a constant.

---
**Note:** The BANDPLOT statement is optimized to work as a Confidence or Prediction band. If the band is self-intersecting (not sorted for X or for Y), then the resulting band is unpredictable. With unsorted data, the band that is generated for an output Raster Image might not match the band that is generated for an output Vector Graphic.

---
**Example: BANDPLOT Statement**
The following graph was generated by the “Example Program” on page 239:
Example Program

Here is the code for this example.

```sas
proc template;
define statgraph bandplot;
begingraph;
  entrytitle "Fit Plot for Weight";
  layout overlay;
    bandplot x=height limitupper=uppermean
      limitlower=lowermean /
      name="band" modelname="fit"
      legendlabel="95% Confidence Limits";
    scatterplot x=height y=weight / primary=true;
    seriesplot x=height y=predict / name="fit"
      legendlabel="Fit Line";
    discretelegend "fit" "band";
  endlayout;
endgraph;
end;
run;

proc sgrender data=sashelp.classfit template=bandplot;
run;
```

### BARCHART Statement

Creates a bar chart computed from input data.
Tips: For charts that have a large number of bars that are very close together, slight variations in spacing that normally occur due to integer rounding can become more obvious. Subpixel rendering provides more precise bar spacing in that case. In SAS 9.4M2 and in earlier releases, specify SUBPIXEL=ON in the BEGINGRAPH statement to enable subpixel rendering. See SUBPIXEL= on page 35. Starting with SAS 9.4M3, subpixel rendering is enabled by default.

To disable subpixel rendering in SAS 9.4M3 and in later releases, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

BARCHART CATEGORY=column | expression <option(s)>;
BARCHART CATEGORY=column | expression
RESPONSE=numeric-column | expression <options>;

Summary of Optional Arguments

Appearance options

BARWIDTH=number
specifies the width of a bar as a ratio of the maximum possible width.

BASELINEATTRS=style-element | (line-options)
specifies the appearance of the baseline.

CLUSTERWIDTH=number
specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

COLORBYFREQ=TRUE | FALSE
specifies whether the bar colors are based on statistical values when the COLORRESPONSE= option is not specified.

COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option or the COLORBYFREQ= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the column or range attribute variable to use to map the bar colors to a continuous color gradient.

CONNECTATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the bar connect lines.

CONNECTBREAK=TRUE | FALSE
specifies whether the connect line is broken for values that have no observations.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the filled bars.

DATATRANSPARENCY=number
specifies the degree of the transparency of the bar fill, bar outline, connect line, and bar labels, if displayed.

DISPLAY=(display-options) | STANDARD | ALL
specifies which bar features to display.

DISPLAYBASELINE=ON | OFF | AUTO
specifies whether the baseline is displayed.

```
DISPLAYZEROLENGTHBAR=TRUE | FALSE
```

specifies whether zero-length bars are drawn.

```
FILLATTRS=style-element | style-element (fill-options) | (fill-options)
```

specifies the appearance of the filled bar area.

```
FILLENDCOLOR=color
```

specifies the end color of the color gradient for the bar fill.

```
FILLPATTERNATTRS=style-element | (fill-pattern-options)
```

specifies the appearance of the pattern-filled areas.

```
FILLTYPE=SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT
```

specifies the bar fill type.

```
INDEX=positive-integer-column | expression
```

specifies indices for mapping bar attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

```
INTERVALBARWIDTH=dimension
```

specifies the width of the bars in an interval bar chart as a ratio of the interval width.

```
ORIENT=VERTICAL | HORIZONTAL
```

specifies the orientation of the Y axis and the bars.

```
OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
```

specifies the appearance of the bar outlines.

```
TARGET=numerical-column | expression
```

specifies the target value for each bar.

**Axes options**

```
BASELINEINTERCEPT=number
```

specifies the response axis intercept for the baseline.

```
PRIMARY=TRUE | FALSE
```

specifies that the data columns for this plot and the plot type be used for determining default axis features.

```
XAXIS=X | X2
```

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

```
YAXIS=Y | Y2
```

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Data tip options**

```
TIP=(role-list) | NONE
```

specifies the information to display when the cursor is positioned over a bar.

```
TIPFORMAT=(role-format-list)
```

specifies display formats for tip columns.

```
TIPLABEL=(role-label-list)
```

specifies display labels for tip columns.

**Label options**

```
BARLABEL=TRUE | FALSE
```

specifies whether the bar statistic value is displayed at the end of the bar.

```
BARLABELATTRS=style-element | style-element (text-options) | (text-options)
```

specifies the text properties of the bar label text.
BARLABELFITPOLICY= AUTO | NONE | INSIDEPREFERRED
specifies a policy for avoiding collisions among the bar labels when labels are displayed.

BARLABELFORMAT= format
specifies the text format used to display the bar label.

LEGENDLABEL= "string"
specifies a label to be used in a discrete legend for this plot.

SEGMENTLABEL= TRUE | FALSE
specifies whether a label is displayed inside each bar segment.

SEGMENTLABELATTRS= style-element | style-element (text-options) | (text-options)
specifies the text properties of the text for the bar segment label.

SEGMENTLABELFITPOLICY= NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

SEGMENTLABELFORMAT= format
specifies the text format for the bar segment labels.

Midpoint options

DISCRETEOFFSET= number
specifies an amount to offset all bars from the category midpoints.

GROUP= column | discrete-attr-var | expression
creates a separate bar segment or bar for each unique group value in the specified column.

GROUP100= NONE | MAGNITUDE | POSITIVE
displays the computed response values (FREQ, SUM, or MEAN), normalized to 100%.

GROUPDISPLAY= STACK | CLUSTER
specifies how to display grouped bars.

GROUPORDER= DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

INCLUDEMISSINGGROUP= TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

ODS options

URL= string-column
specifies an HTML page to display when the bar is selected.

Plot reference options

NAME= "string"
assigns a name to this plot statement for reference in other template statements.

Statistics options

COLORSTAT= FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic to be calculated for the data range of the bar-color gradient.

STAT= FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic to be computed for the Y axis.
Required Arguments

Specifying only CATEGORY= creates a bar chart with bars that, by default, represent frequency counts or percents of CATEGORY. Specifying both CATEGORY= and RESPONSE= creates a bar chart with bars representing summarized values of RESPONSE categorized by CATEGORY.

**CATEGORY=column | expression**
specifies the column or expression for the category values.

**Notes**
You can use X= as an alternative to CATEGORY=. If you use X=, then be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options recognize X as the category role and not as CATEGORY.

For interval category values, if a user-defined format is applied to the category column, the format should map each category value to only one unique formatted value. Otherwise, unexpected results might occur.

**RESPONSE=numeric-column | expression**
specifies the numeric column or expression for the response values.

**Notes**
You can use Y= as an alternative to RESPONSE=. If you use Y=, then be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options will recognize Y as the response role and not RESPONSE in that case.

This option is required only when you want summarized values of RESPONSE that are categorized by CATEGORY.

Optional Arguments

**BARLABEL=TRUE | FALSE**
specifies whether the bar statistic value is displayed at the end of the bar. For grouped clustered bars, each bar is labeled with the summarized value of the bar. For grouped stacked bars, the segmented bar is labeled with the accumulated, summarized value of all the bar segments.

**Default** FALSE

**Note**
By default, the bar-label format is derived from the format that is assigned to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

**Tip**
The font and color attributes for the label are specified by the BARLABELATTRS= option. The text format is specified by the BARLABELFORMAT= option.

**See**
“boolean ” on page 1479 for other Boolean values that you can use.

**BARLABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the text properties of the bar label text.

**Default** The GraphDataText style element.

**Requirement** For this option to take effect, BARLABEL=TRUE must be specified.
BARLABELFITPOLICY=AUTO | NONE | INSIDEPREFERRED

specifies a policy for avoiding collisions among the bar labels when labels are displayed.

AUTO

for a vertical bar chart, rotates the bar labels if the labels exceed the midpoint spacing.

For a horizontal bar chart, always draws the labels horizontally. The following figure shows an example.

![Bar Chart Example](image)

See the BARWIDTH= option for more information about the bar spacing.

NONE

does not rotate the bar labels.

Labels that are too long overlap.

INSIDEPREFERRED

attempts to place the bar labels in a horizontal bar chart inside the bars.

The following figure shows an example.

Note: This feature applies to SAS 9.4M5 and to later releases.
Restrictions

This option applies to non-grouped or cluster-grouped horizontal bar charts only.

Label splitting is not supported when a label is placed inside a bar.

Requirement

For a grouped horizontal bar chart, the GROUPDISPLAY=CLUSTER option must be in effect. Otherwise, the INSIDEPREFERRED policy is ignored.

Interaction

If SEGMENTLABEL=TRUE is in effect, the INSIDEPREFERRED policy is ignored, and the labels are placed outside.

Note

If any bar label cannot fit inside the length of its bar, that label is placed outside of its bar and is fit using the AUTO fit policy.

Labels can collide along their length and along their height. In some cases, if one or more labels collide when the specified fit policy is used, then all of the labels are dropped from the display. When that occurs, the following warning message is written to the SAS log:

WARNING: The bar labels are suppressed. Use BARLABELFITPOLICY=NONE to force the labels to be displayed.

TIP

If the labels collide along their height, then using the BARLABELATTRS= option to reduce the label font size might eliminate the collision.

Default

AUTO

Requirement

For this option to take effect, BARLABEL=TRUE must be specified.
**BARLABELFORMAT** = *format*

specifies the text format used to display the bar label.

**Default**

The bar-label format is derived from the format that is applied to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

**Requirement**

For this option to take effect, BARLABEL=TRUE must be specified.

**Note**

When a bar-label format is specified with this option, the bar labels are formatted as specified by *format*. The specified format is not automatically expanded to accommodate values that are too wide.

**Tip**

If you want the bar-label format to expand automatically for summarized values on the response axis, specify the format for the response column rather than in this option.

---

**BARWIDTH** = *number*

specifies the width of a bar as a ratio of the maximum possible width.

**Default**

0.85

**Range**

0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

**Interaction**

Starting with SAS 9.4M3, the INTERVALBARWIDTH= option overrides this option for an interval bar chart.

**Notes**

This option is needed only to change the default behavior.

By default, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

**Tip**

To remove any inter-bar gap, set BARWIDTH=1.

---

**BASELINEATTRS** = *style-element | (line-options)*

specifies the appearance of the baseline.

**Default**

The GraphAxisLines style element.

**Notes**

The baseline is always drawn by default.

When *style-element* is specified, only the style element’s COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

**Tip**

To suppress the baseline, set the line thickness to 0: `baselineattrs=(thickness=0)`

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Line Options” on page 1522 for available *line-options*.

---

**BASELINEINTERCEPT** = *number*

specifies the response axis intercept for the baseline. Prior to SAS 9.4M5, the baseline is always displayed in the chart, whether for a specified value or for the default value. Starting with SAS 9.4M5, the baseline display is controlled by the
**DISPLAYBASELINE=** option, which is ON by default. When the **BASELINEINTERCEPT=** option is used, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.

![Graph showing a baseline at zero](image)

**Default**

0

**Interactions**

If **GROUPDISPLAY=STACK** is specified, then this option is ignored and the baseline is not displayed.

This option is ignored when the **GROUP100=** option is used.

If necessary, the response axis data range is extended to include the baseline intercept. When a logarithmic response axis is requested and **BASELINEINTERCEPT=** specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set **BASELINEINTERCEPT=** to a positive value.

When **DISPLAYBASELINE=AUTO** is in effect, the baseline is not displayed if the baseline intercept is equal to the minimum or maximum value of the response-axis range.

**Note**

Label positions are automatically adjusted to prevent the labels from overlapping.

**Tips**

Control the appearance of the baseline with the **BASELINEATTRS=** option.

To suppress the baseline prior to **SAS 9.4M5**, use the **BASELINEATTRS=** option to set the line thickness to 0. Starting with **SAS 9.4M5**, specify **DISPLAYBASELINE=OFF**.

The baseline does not add a tick or a tick value to the axis. To label the baseline, use a **REFERENCELINE** statement to overlay a line with the same X or Y value and include the **CURVELABEL=** option to specify the label text.

**CLUSTERWIDTH=** *number*

specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.
Default: 0.85

Range: 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Requirement: For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

Interaction: When GROUPDISPLAY=CLUSTER, the default BARWIDTH is 1.0.

**COLORBYFREQ=TRUE | FALSE**

specifies whether the bar colors are based on statistical values when the COLORRESPONSE= option is not specified. Setting this option to TRUE enables you to color the bars based on frequency counts, percentages, or proportions.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default: FALSE

Requirement: The COLORSTAT= option must be FREQ, PCT, or PROPORTION for this option to have any effect.

Interaction: This option is ignored when the COLORRESPONSE= option is specified.

When the GROUP= option is specified with the COLORBYFREQ= option, the color attributes are controlled by the COLORBYFREQ= option.

The COLOR= suboption of the FILLATTRS=, FILLPATTERNATTRS=, and OUTLINEATTRS= options overrides this option for the associated color attribute.

Note: This option is independent of the STAT= and RESPONSE= options.
Tips

Use the **COLORSTAT=** option to specify whether frequency counts, percentages, or proportions are computed for the **COLORRESPONSE=** column.

Use the **FILLTYPE=** option to specify whether each bar is filled with a solid color or with a gradient color.

See

“Example 4: Bar Chart with Bar Colors Controlled by a Statistic” on page 278

**COLORMODEL=**color-ramp-style-element | (color-list)

specifies a color ramp to use with the **COLORRESPONSE=** option or the **COLORBYFREQ=** option.

**color-ramp-style-element**

specifies the name of a color-ramp style element.

The style element should contain these style attributes:

**STARTCOLOR**

specifies the color for the smallest data value of the **COLORRESPONSE=** column.

**NEUTRALCOLOR**

specifies the color for the midpoint of the range of the **COLORRESPONSE=** column.

**ENDCOLOR**

specifies the color for the highest data value of the **COLORRESPONSE=** column.

**color-list**

specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement**

The list of colors must be enclosed in parentheses.

See

“color ” on page 1480

**Defaults**

For outline-only bars, the ThreeColorAltRamp style element

For bars with fill, the ThreeColorRamp style element

**Interaction**

For this option to take effect, the **COLORRESPONSE=** option or the **COLORBYFREQ=TRUE** option must also be specified.

**Tip**

Use the **DISPLAY=** option to specify whether outlines and fills are displayed.

**COLORRESPONSE=**numeric-column | range-attr-var | expression

specifies the column or range attribute variable to use to map the bar colors to a continuous color gradient.

**Note:** This feature applies to **SAS 9.4M3** and to later releases.
range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

"RANGEATTRVAR Statement"

Restriction
A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See
"RANGEATTRVAR Statement" on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. Each bar is colored using one color from the gradient range. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Requirement
For a grouped plot, the COLORRESPONSE values should remain constant for each group value. If the COLORRESPONSE column has multiple values for a single GROUP value, unexpected results might occur.

Interactions
The COLORBYFREQ= option is ignored when this option is specified.

When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

When fill, fill pattern, or both are displayed, this option overrides suboption COLOR= in the FILLATTRS= option and in the FILLPATTERNATTRS= option and varies the color according to the color gradient or the attribute map.

When only the outlines are displayed, this option overrides suboption COLOR= in the OUTLINEATTRS= option and varies the outline color according to the color gradient or the attribute map.

Tips
To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

Use the COLORSTAT= option to specify the statistic to compute for the COLORRESPONSE= column.

Use the FILLTYPE= option to specify whether each bar is filled with a solid color or with a gradient color.

For a numeric column or expression, the ThreeColorRamp style element defines the fill color gradient, and the ThreeColorAltRamp style element defines the outline color gradient.

COLORSTAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic to be calculated for the data range of the bar-color gradient.

Note: This feature applies to SAS 9.4M3 and to later releases.
The statistics that are available depend on the COLORRESPONSE= and COLORBYFREQ= option specifications. When the COLORRESPONSE= option is specified, the following values are valid:

- **SUM**
- **MEAN**

When the COLORRESPONSE= option is not specified and COLORBYFREQ=TRUE is in effect, the following values are valid:

- **FREQ** frequency count
- **PCT** percentages between 0 and 100
- **PROPORTION** proportions between 0 and 1

**Defaults**

- FREQ when the COLORRESPONSE= option is not specified and COLORBYFREQ=TRUE is in effect.
- SUM when the COLORRESPONSE= option is specified.

**Interactions**

This option is ignored when the COLORRESPONSE= option is not specified and COLORBYFREQ=FALSE is in effect.

This option might affect existing SAS programs. For programs written before SAS 9.4M3, if STAT= and COLORRESPONSE= are specified in a BARCHART statement, then the bar-chart colors and color statistic might change from those of the previous SAS releases. To restore the original colors and color statistic in that case, set COLORSTAT= in the BARCHART statement to the same statistic that is specified in STAT=.

**Note**

This option is independent of the STAT= and RESPONSE= options.

**See**

- COLORBYFREQ= on page 248
- COLORRESPONSE= on page 249
- STAT= on page 268

“Example 4: Bar Chart with Bar Colors Controlled by a Statistic” on page 278

**CONNECTATTRS=**

```
CONNECTATTRS=style-element | style-element (line-options) | (line-options)
```

specifies the appearance of the bar connect lines.

**Default**

The GraphConnectLine style element.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**CONNECTBREAK=**

```
CONNECTBREAK=TRUE | FALSE
```

specifies whether the connect line is broken for values that have no observations.

**Note:** This feature applies to SAS 9.4M5 and to later releases.
Default: FALSE

Requirement: DISPLAY= must include CONNECT for this option to have any effect.

Interaction: This option is ignored when the GROUP= option is in effect.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**

enhances the visual appearance of the filled bars. The following figure shows bars with each of the skins applied.

<table>
<thead>
<tr>
<th>NOIE</th>
<th>CRISP</th>
<th>GLOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATTE</td>
<td>PRESSED</td>
<td>SHEEN</td>
</tr>
</tbody>
</table>

Default: The **DATASKIN=** option value that is specified in the BEGINGRAPH statement. If not specified, then the GraphSkins:DataSkin style element value is used.

Restriction: Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

Requirement: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Interactions: This option overrides the BEGINGRAPH statement **DATASKIN=** option.

The data skin appearance is based on the **FILLATTRS=** color.

When a data skin is applied, all bar outlines are set by the skin, and the **OUTLINEATTRS=** option is ignored.

When **FILLTYPE=**GRADIENT is in effect, DATASKIN=SHEEN is ignored. In that case, use one of the other skins.

**DATATRANSPARENCY=**number

specifies the degree of the transparency of the bar fill, bar outline, connect line, and bar labels, if displayed.
The **FILLATTRS=** option can be used to set transparency for just the filled bar area. You can combine this option with **FILLATTRS=** to set one transparency for the bar outlines and connect lines but a different transparency for the bar fills. Example:

```
data transparency=0.2 fillattrs=(transparency=0.6)
```

**DISCRETEOFFSET=number**

specifies an amount to offset all bars from the category midpoints.

**Default** 0 (no offset, all bars are centered on the category midpoints)

**Range** -0.5 to +0.5, where 0.5 represents half the distance between category ticks. Normally, a positive offset is to the right when **ORIENT=VERTICAL**, and up when **ORIENT=HORIZONTAL**. (If the layout's axis options set **REVERSE=TRUE**, then the offset direction is also reversed.)

**Tip** Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the **OFFSETMIN=** and **OFFSETMAX=** axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**See** "About the DISCRETEOFFSET= Option" on page 272

"About the DISCRETEOFFSET= Option" on page 272 for information about the **REVERSE=**, **OFFSETMIN=**, and **OFFSETMAX=** axis options

**ORIENT=**

**DISPLAY=(display-options) | STANDARD | ALL**

specifies which bar features to display.

**display-options**

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**CONNECT**

displays the connect lines, which are line segments that connect adjacent midpoints at the end of each bar.

**FILL**

displays the bar fill color

**FILLPATTERN**

displays the bar fill pattern

**Tip** For this option to have any effect, the active ODS style must specify a fill pattern or a fill pattern must be specified with the **FILLPATTERNATTRS=** option.

**OUTLINE**

displays the bar outline
**STANDARD**

specifies FILL and OUTLINE

**ALL**

specifies all features: CONNECT, FILL, FILLPATTERN, and OUTLINE

Default  The GraphBar:DisplayOpts style reference.

Interaction  Connect lines are not drawn when the GROUP= option is in effect.

Note  The connect lines are drawn in axis order starting with SAS 9.4M3. They are drawn in data order in prior releases.

Tips  Use the OUTLINEATTRS=, FILLATTRS=, and FILLPATTERNATTRS= options to control the appearance of the bars.

Use CONNECTATTRS= to control the appearance of the connect lines.

You can specify both FILL and FILLPATTERN to combine solid fills and pattern fills in the bars.

**DISPLAYBASELINE=ON | OFF | AUTO**

specifies whether the baseline is displayed.

Note: This feature applies to SAS 9.4M5 and to later releases.

**ON**

always displays the baseline.

**OFF**

does not display the baseline.

**AUTO**

displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

This is the typical case when the bar chart includes both positive and negative bars, and the default baseline intercept of 0 is used. If the baseline intercept is equal to the minimum or maximum value of the response-axis range, the baseline is not displayed.

Default  ON

**Tip**  By default, the baseline intercept is 0. Use the BASELINEINTERCEPT= option to change the baseline intercept.

**DISPLAYZEROLENGTHBAR=TRUE | FALSE**

specifies whether zero-length bars are drawn.

Note: This feature applies to SAS 9.4M3 and to later releases.

A zero-length bar is displayed as a line spanning the normal bar width at the bar-chart baseline on the response axis. When this option is set to TRUE, zero-length bars are displayed. Otherwise, they are suppressed. The following figure shows a simple example of each outcome. In the figure, the plot wall outline, category axis line, and bar-chart baseline are suppressed for clarity.
FILLATTRS=\textit{style-element} | \textit{style-element} (\textit{fill-options}) | (\textit{fill-options})
specifies the appearance of the filled bar area.

\textbf{Defaults}
For non-grouped data, the GraphDataDefault:Color style reference
For grouped data, the Color attribute of the GraphData1–GraphDataN style elements.

\textbf{Interaction}
When \texttt{COLORRESPONSE=} is in effect and the \texttt{DISPLAY=} option enables FILL display, the FILLATTRS= suboption \texttt{COLOR=} is ignored, and the bar fill colors vary according to the gradient.

\textbf{Tip}
The \texttt{DATATRANSPARENCY=} option sets the transparency for the bar fills, bar outlines, and connect lines. You can combine this option with \texttt{DATATRANSPARENCY=} to set one transparency for the bar outlines and connect lines but a different transparency for the bar fills. Example:

\begin{verbatim}
datatransparency=0.2 fillattrs=(transparency=0.6)
\end{verbatim}

\textbf{See}
“General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}.

“Fill Color Options” on page 1520 for available \textit{fill-options}.

FILLENDCOLOR=\textit{color}
specifies the end color of the color gradient for the bar fill.

\textbf{Note:} This feature applies to SAS 9.4M5 and to later releases.

\textbf{Default}
the graph wall background color, even when the wall is not displayed

\textbf{Requirement}
\texttt{FILLTYPE=COLORGRADIENT} must be in effect. Otherwise, this option is ignored.

\textbf{See}
“\textit{color} ” on page 1480
**FILLPATTERNATTRS=** `style-element | (fill-pattern-options)`  
specifies the appearance of the pattern-filled areas.

**Defaults**  
For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

**Restriction**  
In SAS 9.4M4 and in earlier releases, the only styles that are delivered by SAS that support fill patterns are JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER. If `style-element` is specified and the active ODS style does not support fill patterns, this option is ignored.

**Interactions**  
For this option to take effect, the `DISPLAY=` option must include FILLPATTERN among the display options.

When `COLORRESPONSE=` is in effect and the `DISPLAY=` option enables FILLPATTERN display, the FILLPATTERNATTRS= suboption COLOR= is ignored, and the fill-pattern colors vary according to the gradient.

**Tip**  
In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User's Guide.

**See**  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Fill Pattern Options” on page 1521 for available `fill-pattern-options`.

**FILLTYPE=** `SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT`  
specifies the bar fill type.

---

**Note:** This feature applies to SAS 9.4M2 and to later releases.

---

**SOLID**  
fills each bar with the color that is assigned to that bar.

**ALPHAGRADIENT**  
fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.

---

**Note:** This feature applies to SAS 9.4M5 and to later releases.

---

**Tips**  
The fill color is determined by a style element or by the FILLATTRS= option COLOR= suboption. The initial transparency is determined by the DATATRANSPARENCY= option or by the FILLATTRS= option TRANSPARENCY= suboption.
For grouped plots, use the FILLATRRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**COLORGRADIENT**

fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the FILLENDCOLOR= option.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

<table>
<thead>
<tr>
<th>Alias</th>
<th>GRADIENT</th>
</tr>
</thead>
</table>

**Interaction**
The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

**Tips**
Use the DATATRANSPARENCY= option or the FILLATRRS= option TRANSPARENCY= suboption to set the initial transparency in the gradients.

For grouped plots, use the FILLATRRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**See**
“FILLENDCOLOR=color” on page 255

**GRADIENT**

fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.

**Note:** This option applies to SAS 9.4M2 through SAS 9.4M4. Starting with SAS 9.4M5, ALPHAGRADIENT replaces GRADIENT, and GRADIENT is changed to an alias of COLORGRADIENT.

**Interactions**
The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

In SAS 9.4M2, FILLTYPE=GRADIENT is ignored when GROUPDISPLAY=STACK is in effect. Starting with SAS 9.4M3, FILLTYPE=COLORGRADIENT is honored in that case.

**Tips**
The initial fill color is determined by a style element or by the FILLATRRS= option COLOR= suboption. The initial transparency is determined by the DATATRANSPARENCY= option or by the FILLATRRS= option TRANSPARENCY= suboption.

For grouped plots, use the FILLATRRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**Default**
SOLID
Interaction  The DISPLAY= option must include FILL for this option to have any effect.

Note  The output for SAS programs written prior to SAS 9.4M5 that specify FILLTYPE=GRADIENT might change starting with SAS 9.4M5. To restore the original appearance, specify FILLTYPE=ALPHAGRADIENT.

See  DATASKIN= on page 252

GROUP=column | discrete-attr-var | expression  creates a separate bar segment or bar for each unique group value in the specified column.

discrete-attr-var  specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction  A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

The bar display depends on the setting for the GROUPDISPLAY= option. For example, for a vertical bar chart with GROUPDISPLAY=STACK, bar segments are stacked to form the bar. The height of each segment represents the corresponding group value’s proportional contribution to the response value.

A distinct bar or bar segment is created for each group value by varying the visual attributes of the bar display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern. If the active ODS style does not specify FillPattern, fill patterns are not displayed by default.</td>
</tr>
<tr>
<td>Display Feature</td>
<td>Style Attributes That Control Default Group Appearance</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Outline        | - if the outline is enabled with fill color, fill pattern, or both, ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style  
|                | - if the outline is the only display feature enabled, ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element |

1 The default display features are determined by the DisplayOpts attribute of the GraphBar style element.

Interactions

Connect lines are not drawn for grouped data.

By default, the group values are mapped in the order of the data. Use the GROUPORDER= option to control the sorting order of the grouped bar segments. Use the INDEX= option to alter the default sequence of colors and line patterns.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

When both the GROUP= and COLORRESPONSE= options are specified, the color attributes are controlled by the COLORRESPONSE= option.

Notes

The bar display depends on the setting for the GROUPDISPLAY= option.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

Tip

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the bar outlines, but you can use the PATTERN= setting on the OUTLINEATTRS= option to assign the same line pattern to all bar outlines and connect lines.

See

“Example 2: Grouped Vertical Bar Chart” on page 275

“DISCRETEATTRVAR Statement” on page 1435

GROUP100=NONE | MAGNITUDE | POSITIVE displays the computed response values (FREQ, SUM, or MEAN), normalized to 100%.
Note: This feature applies to SAS 9.4M3 and to later releases.

NONE
displays the summarized data.

MAGNITUDE
normalizes both the negative and positive values to 100% by magnitude, and displays the group values, preserving the sign.

The positive values are displayed above the bars for a vertical bar chart and on the right end for a horizontal bar chart. The negative values are displayed enclosed in parentheses below the bars for a vertical bar chart and on the left end for a horizontal bar chart.

The following figure illustrates the effect of MAGNITUDE on stacked bars in a vertical bar chart.

![Regional Profits and Losses by Quarter](chart1)

POSITIVE
drops the negative values and normalizes only the positive values to 100%.

The following figure demonstrates the effect of POSITIVE on clustered bars in a vertical bar chart. This chart uses the same data as the chart in the previous figure.

![Regional Profits by Quarter](chart2)

Notice that the negative values are dropped from the chart.
Default    NONE  

Requirement  The GROUP= option must be specified for this option to have any effect.

Interaction  When this option is used, the BASELINEINTERCEPT= and TARGET= options are ignored.

Note  You can use this option with any value for the GROUPDISPLAY= option.

Tip  To display the values, specify BARLABEL=TRUE.

**GROUPDISPLAY=STACK | CLUSTER**  

specifies how to display grouped bars.

**STACK**

displays group values as stacked segments within the category bar.

**CLUSTER**

displays group values as separate adjacent bars that replace the single category bar.

Each cluster of group values is centered at the category midpoint on the axis. This example illustrates the clusters and also how groups are displayed when they have an unequal number of unique values.
When you use the `BARLABEL=` option and the `GROUP=` option, the `BARLABEL` values are displayed for each bar when `GROUPDISPLAY=CLUSTER`. When `GROUPDISPLAY=STACK`, the whole bar is labeled at the top.

When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

For a linear response axis, when `STAT=MEAN` or `STAT=PCT`, the axis tick values might be displayed as integer values when `GROUPDISPLAY=STACK`. Changing `GROUPDISPLAY=` to `CLUSTER` in that case might cause the axis values to change to decimal values. To keep the integer axis values in both cases, you can specify the `INTEGER=TRUE` option for the response axis. See `INTEGER=` on page 1015.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING

specifies the ordering of the groups within a category.

- **DATA**
  orders the groups within a category in the group-column data order.

- **REVERSEDATA**
  orders the groups within a category in the reverse group-column data order.

  **Note:** This feature applies to SAS 9.4M2 and to later releases.

  **Tip** This option is useful when you want to reverse the category axis.

- **ASCENDING**
  orders the groups within a category in ascending order.

- **DESCENDING**
  orders the groups within a category in descending order.

**Defaults**

- DATA if the data is a SAS data set.
- ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

**Interactions**

This option is ignored if the `GROUP=` option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

**Notes**

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position.
purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

<table>
<thead>
<tr>
<th>INCLUDEMISSINGGROUP=</th>
<th>TRUE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies whether missing values of the group variable are included in the plot.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default</td>
<td>TRUE</td>
<td></td>
</tr>
<tr>
<td>Interaction</td>
<td>For this option to take effect, the GROUP= option must also be specified.</td>
<td></td>
</tr>
<tr>
<td>Tip</td>
<td>The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.</td>
<td></td>
</tr>
<tr>
<td>See</td>
<td>“boolean ” on page 1479 for other Boolean values that you can use.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INDEX=positive-integer-column</th>
<th>expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies indices for mapping bar attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td>The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.</td>
</tr>
<tr>
<td></td>
<td>The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.</td>
</tr>
<tr>
<td></td>
<td>All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.</td>
</tr>
<tr>
<td>Interaction</td>
<td>For this option to take effect, the GROUP= option must also be specified.</td>
</tr>
<tr>
<td>Notes</td>
<td>The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.</td>
</tr>
<tr>
<td></td>
<td>If you do not use this option, then the group values are mapped in the order of the data.</td>
</tr>
</tbody>
</table>
**Tip**
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**INTERVALBARWIDTH=** *dimension*

specifies the width of the bars in an interval bar chart as a ratio of the interval width.

---

**Note:** This feature applies to SAS 9.4M3 and to later releases.

<table>
<thead>
<tr>
<th>Default</th>
<th>The width specified by the BARWIDTH= option.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restriction</strong></td>
<td>This option applies only to an interval category axis. When the category axis is discrete, this option is ignored.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>When the category data is interval, this option overrides the BARWIDTH= option.</td>
</tr>
<tr>
<td><strong>Tips</strong></td>
<td>To make the category axis type linear or time, include TYPE=LINEAR or TYPE=TIME in the category axis options or assign the role of primary plot to a plot that makes the category axis linear or time.</td>
</tr>
<tr>
<td></td>
<td>The bar width that you specify with this option is honored even if the bars overlap. If the bars overlap, reduce the interval bar width or use the BARWIDTH= option instead.</td>
</tr>
</tbody>
</table>

See “dimension” on page 1480

**LEGENDLABEL=** *"string"*

specifies a label to be used in a discrete legend for this plot.

<table>
<thead>
<tr>
<th>Default</th>
<th>The response-variable label. If a label is not defined, then the response-variable name is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restriction</strong></td>
<td>This option applies only to an associated DISCRETELEGEND statement.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>If the GROUP= option is specified, then this option is ignored.</td>
</tr>
</tbody>
</table>

**NAME=** *"string"*

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

<table>
<thead>
<tr>
<th>Restrictions</th>
<th>The string is case sensitive, cannot contain spaces, and must define a unique name within the template.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>The string is used as the default legend label if the LEGENDLABEL= option is not used.</td>
</tr>
</tbody>
</table>

**ORIENT=** VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.
When this option is set to HORIZONTAL, the category variable appears on the Y (or Y2) axis and the response variable appears on the X (or X2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

When this option is set to VERTICAL, the category variable appears on the X (or X2) axis and the response variable appears on the Y (or Y2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

If you change the orientation of the bar chart, then you should adjust the layout container’s axis options appropriately.

**OUTLINEATTRS=** *style-element | style-element (line-options) | (line-options)*

specifies the appearance of the bar outlines.

**Defaults**

For non-grouped data, the ContrastColor, LineThickness, and LineStyle attributes of the GraphOutlines style element.

For grouped data and filled bars, the ContrastColor attribute of the GraphData1–GraphDataN style elements, and the LineThickness and LineStyle attributes of the GraphOutlines style element.

For grouped data and unfilled bars, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the LineThickness attribute of the GraphOutlines style element.

**Interactions**

For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

If the DATASKIN= option applies a data skin, then this option is ignored.

When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the bar outline colors vary according to the gradient.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Line Options” on page 1522 for available *line-options*.

**PRIMARY=TRUE | FALSE**

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**

FALSE

**Restriction**

This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note**

In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same
axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979
“boolean ” on page 1479 for other Boolean values that you can use.

SEGMENTLABEL=TRUE | FALSE
specifies whether a label is displayed inside each bar segment.

Note: This feature applies to SAS 9.4M2 and to later releases.

For an ungrouped bar chart or for a grouped bar chart with GROUPDISPLAY=CLUSTER, AUTO displays a bar label inside each bar. The label displays the statistic for that bar. For a grouped bar chart with GROUPDISPLAY=STACK, AUTO displays a label inside each bar segment. Each segment label displays the statistic for that bar segment, as shown in the following figure.

When this value is set to FALSE, no labels are displayed inside the bars.

Default FALSE

Note By default, the segment-label format is derived from the format that is assigned to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Tips For a grouped bar chart with GROUPDISPLAY=STACK, specify both SEGMENTLABEL=TRUE and BARLABEL=TRUE to display a label for each bar segment and a label for the entire bar.

Use the SEGMENTLABELATTRS= option to modify the appearance of the label text.

Use the SEGMENTLABELFITPOLICY= option to specify a policy for fitting the labels inside the bars.
Use the **SEGMENTLABELFORMAT** option to modify the format of the segment labels.

See “**boolean**” on page 1479 for other Boolean values that you can use.

**SEGMENTLABELATTRS=** *style-element | style-element (text-options) | (text-options)*

specifies the text properties of the text for the bar segment label.

- **Note:** This feature applies to **SAS 9.4M2** and to later releases.

- **Default:** The GraphDataText style element.

- **Interaction:** This option is ignored when **SEGMENTLABEL=FALSE**.

- **See** “**General Syntax for Attribute Options**” on page 1519 for the syntax for using a *style-element*.

- **See** “**Text Options**” on page 1525 for available *text-options*.

**SEGMENTLABELFITPOLICY=** *NONE | NOCLIP | THIN*

specifies a policy for fitting the bar segment labels within the bar segments.

- **Note:** This feature applies to **SAS 9.4M2** and to later releases.

- **NONE**
  
  no attempt is made to fit each segment label within its bar. Long bar segment labels might overlap other graphical elements.
  
  The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.

- **NOCLIP**
  
  does not clip bar segment labels that extend beyond the plot area.
  
  Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.

- **THIN**
  
  drops any bar segment label that does not fit within its segment.
  
  For a vertical bar chart, the label width must not exceed the bar width, and the text height must not exceed the segment height. For a horizontal bar chart, the label text height must not exceed the bar width, and the label length must not exceed the segment length.

- **Default:** **THIN**

- **Interaction:** This option is ignored when **SEGMENTLABEL=FALSE**.

**SEGMENTLABELFORMAT=** *format*

specifies the text format for the bar segment labels.

- **Note:** This feature applies to **SAS 9.4M2** and to later releases.
The segment-label format is derived from the format that is assigned to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

Interaction
This option is ignored when SEGMENTLABEL=FALSE.

Note
When a segment-label format is specified with this option, the segment labels are formatted as specified by format. The specified format is not automatically expanded to accommodate values that are too wide.

Tip
If you want the segment-label format to expand automatically for the summarized values on the response axis, specify the format for the response column rather than in this option.

STAT=FREQ | PCT | SUM | MEAN | PROPORTION
specifies the statistic to be computed for the Y axis. For bar charts with no RESPONSE= column:

FREQ frequency count
PCT percentages between 0 and 100
PROPORTION proportions between 0 and 1

Note: Prior to SAS 9.4, PCT displayed proportions between 0 and 1. To restore the original PCT results in SAS 9.4 and later releases, specify PROPORTION instead.

For bar charts with a RESPONSE= column:

SUM
MEAN

Defaults
For bar charts with no RESPONSE= column, the default is FREQ.
For bar charts with a RESPONSE= column, the default is SUM.

Note
When this option is used with the GROUP=group option, the specified statistic is computed for each segment that is created for the unique group values.

Tip
If this option is used with COLORRESPONSE= in SAS programs that were written before SAS 9.4M3, the bar-chart colors and color statistic might change from those of the previous SAS releases. To restore the original colors and color statistic, set COLORSTAT= in the BARCHART statement to the same statistic that is specified in STAT=.

TARGET=numeric-column | expression
specifies the target value for each bar. The visual representation is a triangle with a line at the target value.

```plaintext
layout overlay;
  barchart category=type response=mpg_highway / barwidth=.8
    target=mpg_city group=origin groupdisplay=cluster
  name='bar';
```
discretelegend 'bar';
endlayout;

layout overlay;
barchartparm category=type response=mpg_highway / barwidth=.8
   target=mpg_city group=origin groupdisplay=cluster
   name='bar';
discretelegend 'bar';
endlayout;

Target Markers Show City Mileage

<table>
<thead>
<tr>
<th>Type</th>
<th>Hybrid</th>
<th>Truck</th>
<th>Wagon</th>
<th>Sports</th>
<th>Sedan</th>
<th>SUV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>Asia</td>
<td>Europe</td>
<td>USA</td>
<td>Asia</td>
<td>Europe</td>
<td>USA</td>
</tr>
</tbody>
</table>

Default
No targets are displayed.

Interactions
For this option to take effect, the RESPONSE= argument must also be used.

If the GROUP= option is used and GROUPDISPLAY= STACK, then this option is ignored.

This option is ignored when the GROUP100= option is used.

Tips
The statistic indicated by the STAT= option applies to the TARGET= column. If a constant value is desired for each target, then specify it only once for repeated category (X) values (or category and GROUP combinations), and leave the other target values missing.

The target color is that of the bar outline.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a bar. If this option is used, then the information specified replaces all of the information that is displayed by default.

(role-list)
an ordered, space-separated list of unique BARCHART roles. BARCHART roles include CATEGORY or X, RESPONSE or Y, COLORRESPONSE, INDEX, GROUP, and TARGET.

Notes
For the category and response roles, the TIP= option recognizes only the category and response arguments that you use in the BARCHART statement. If you use the CATEGORY= and RESPONSE= arguments, then you must specify roles CATEGORY
and RESPONSE. Conversely, if you use the X= and Y= arguments, then you must specify roles X and Y.

The COLORRESPONSE role is valid starting with **SAS 9.4M3**.

**Example**
The following example displays data tips for the columns assigned only to the roles CATEGORY and RESPONSE:

```
TIP=(CATEGORY RESPONSE)
```

**NONE**
suppresses data tips and URLs (if requested) from the plot.

**Default**
The columns assigned to these roles are automatically included in the data tip information: CATEGORY or X, RESPONSE or Y, COLORRESPONSE, and GROUP.

**Requirement**
To generate data tips in the output, you must include an ODS GRAPHICS ON statement with the IMAGEMAP option specified, and you must write the output to the ODS HTML destination.

**Interaction**
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tip**
You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

**TIPFORMAT=**(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

```
(role-format-list)
```
a space-separated list of role-name = format pairs.

**Example**
```
TIP=(RESPONSE)
TIPFORMAT=(RESPONSE=DOLLAR12.)
```

**Default**
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles.

**TIPLABEL=**(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

```
role-label-list
```
a space-separated list of rolename ="string" pairs.

**Example**
```
TIP=(RESPONSE)
TIPLABEL=(RESPONSE="Average Sales")
```

**Default**
The column label or column name of the column assigned to the role.
Restriction  Only the roles that appear in the TIP= option are used.

Requirement  A column must be assigned to each of the specified roles.

**URL=string-column**  
specifies an HTML page to display when the bar is selected.

*string-column*  
specifies a column that contains a valid HTML page reference (HREF) for each bar that is to have an active link.

**Example**  
http://www.sas.com

Requirement  To generate selectable bars, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interactions  This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Notes  For non-grouped data, the values of the column are expected to be same for each unique category value. If they are not, then the results might be unpredictable.

For grouped data, the values of the column are expected to be the same for each unique category and GROUP combination.

Tips  The URL value can be blank for some category values, meaning that no action is taken when the bars for those category values are selected.

The URL value can be the same for different category values, meaning that the same action is taken when the bars for those category values are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**  
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default  X

Interaction  The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**  
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.
Default: Y

```
Interactions
This option is ignored if the RESPONSE= argument is not specified.
```

The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

Details

Statement Description

The BARCHART statement creates a bar chart with bars that represent summarized response values. The response values are categorized by the unique category values or, starting with SAS 9.4M3, by the bins in binned category data. The BARCHART statement takes nonsummarized data as input and calculates the appropriate summarization statistics (sum, mean, and so on) for each unique category value or category bin. Prior to SAS 9.4M3, the category axis for a bar chart must be discrete. Starting with SAS 9.4M3, the category axis can be discrete, linear, or time. The response axis in all cases is interval.

When the chart is oriented vertically, the X (or X2) axis is used for CATEGORY and the Y (or Y2) axis is used for RESPONSE. When it is oriented horizontally, the X (or X2) axis is used for RESPONSE and the Y (or Y2) axis is used for CATEGORY. (See ORIENT= on page 264.)

By default, if the category column is character, then the bars in the chart appear in the order in which the category values are present in the input data. If the category column is numeric, then the values are presented in ascending order. For non-grouped data, duplicated category values are summarized into a unique value. For grouped data, the category values are summarized as needed. (See the GROUP= option.)

Starting with SAS 9.4M3, for numeric category values, an interval bar chart is generated only when the category axis type is linear or time. To specify a category axis type of linear or time, include the TYPE= option in the category axis options, or assign the role of primary plot to a plot that sets the category axis type to linear or time automatically. By default, a bar is drawn for each unique category value, which can result in a large number of bars for numeric category data.

When binning is used, for each bin, a summarization statistic is computed, and a bar is drawn that represents that statistic. The width of each bar spans the width of the bin that it represents. The left-most edge of the bar represents the start of the bin, and the right-most edge represents the end. See "Example 1: Horizontal Bar Chart" on page 275.

**TIP** Prior to SAS 9.4M3, use the HISTOGRAM statement to create a bar chart that represents response values along an interval axis.

About the DISCRETEOFFSET= Option

The DISCRETEOFFSET= option is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BARCHART statements are used with different response variables, then the bars for matching category values are centered on the midpoints and the bars are superimposed. To make it easier to distinguish among superimposed bars, you can
assign a different BARWIDTH= setting to each BARCHART statement in the overlay:

```sas
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

barchart category=year response=A_revenue / stat=sum name="A"
    legendlabel="A" barwidth=0.8;
barchart category=year response=B_revenue / stat=sum name="B"
    legendlabel="B" barwidth=0.6;
barchart category=year response=C_revenue / stat=sum name="C"
    legendlabel="C" barwidth=0.4;

discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```

To place the different response values side by side, you can assign a different offset to each BARCHART statement. The BARWIDTH= option can be used with DISCRETEOFFSET= to create narrower bars that require less width within the plot area:

```sas
layout overlay / cycleattrs=true
    xaxisopts=(display=(tickvalues))
    yaxisopts=(label="Revenue" offsetmax=0.2);

barchart category=year response=A_revenue / stat=sum name="A"
    legendlabel="A" discreteoffset=-0.3 barwidth=0.3;
barchart category=year response=B_revenue / stat=sum name="B"
    legendlabel="B" discreteoffset=0 barwidth=0.3;
barchart category=year response=C_revenue / stat=sum name="C"
    legendlabel="C" discreteoffset=+0.3 barwidth=0.3;

discretelegend "A" "B" "C" / title="Product:"
    location=inside halign=right valign=top;
endlayout;
```
Different combinations of DISCRETEOFFSET and BARWIDTH can be used to get the effect that you want. Gaps can be created between bars by providing a narrower bar width. Or, bars can be overlapped if the bar widths are increased in proportion to the discrete offset.

```
layout overlay / cycleattrs=true
  xaxisopts=(display=(tickvalues))
  yaxisopts=(label="Revenue" offsetmax=0.2);

barchart category=year response=A_revenue / stat=sum name="A"
  legendlabel="A" datatransparency=0.2
  discreteoffset=-0.2 barwidth=0.5 ;

barchart category=year response=B_revenue / stat=sum name="B"
  legendlabel="B" datatransparency=0.2
  discreteoffset=0   barwidth=0.5 ;

barchart category=year response=C_revenue / stat=sum name="C"
  legendlabel="C" datatransparency=0.2
  discreteoffset=+0.2 barwidth=0.5 ;

discretelegend "A" "B" "C" / title="Product:"
   location=inside halign=right valign=top;
endlayout;
```
Examples:

Example 1: Horizontal Bar Chart
The following graph was generated by the “Example Program” on page 275:

```
proc template;
  define statgraph barchart;
  begingraph;
    entrytitle "Average Mileage by Vehicle Type";
    layout overlay;
      barchart category=type response=mpg_highway / stat=mean orient=horizontal;
    endlayout;
  endgraph;
end;

proc sgrender data=sashelp.cars template=barchart;
run;
```

Example 2: Grouped Vertical Bar Chart
The following graph was generated by the “Example Program” on page 276:
Example Program

```
proc template;
  define statgraph barchart;
    begingraph / attrpriority=none;
      entrytitle "Average MPG by Vehicle Type and Origin";
      layout overlay;
        barchart x=type y=mpg_highway / name="meanmpg"
          stat=mean display=all
          group=origin groupdisplay=cluster
          barlabel=true barlabelformat=5.1;
        discretelegend "meanmpg";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.cars template=barchart;
run;
```

Example 3: Interval Bar Chart

Interval bar charts are available starting with SAS 9.4M3. In SAS 9.4M2 and in earlier releases, use the HISTOGRAM statement to generate an interval bar chart. The following graph was generated by the “Example Program” on page 277:
Example Program

Here is the SAS code.

```sas
proc template;
   define statgraph cylinders;
   begingraph;
      entrytitle "Interval Bar Chart of Vehicle Engine Cylinders";
      layout overlay / xaxisopts=(label="Engine Cylinders" type=linear
                     lineattrs=(tickvaluelist=(3 4 5 6 8 10 12)))
      yaxisopts=(label="Percentage of Vehicles Manufactured"
                     griddisplay=on linearopts=(tickvalueformat=percent7.1));
      barchart category=cylinders / stat=proportion
                     barlabel=true barlabelformat=percent7.1;
      endlayout;
   endgraph;
end;
run;

proc sgrender data=sashelp.cars template=cylinders;
run;
```

Details

An interval bar chart can be generated only when the category axis type is LINEAR or TIME. In this example, the TYPE=LINEAR option is included in the XAXISOPTS= options. With numeric category data, a bar is drawn for each unique category value. In some cases, that can generate too many bars in the resulting chart. In this example, there are only seven unique values. The TICKVALUELIST= option is used in the XAXISOPTS= option to display all of the values on the category axis.
Example 4: Bar Chart with Bar Colors Controlled by a Statistic

The ability to use a computed statistic to control the bar colors in a bar chart is available starting with SAS 9.4M3. This example uses the COLORBYFREQ=TRUE option to enable a computed statistic to control the bar colors and the COLOSTAT=PCT to specify percentage as the controlling statistic. Here is the output from “Example Program” on page 278.

Example Program

```sas
proc template;
  define statgraph barchart;
  begingraph;
    entrytitle "Average Mileage by Vehicle Type";
    layout overlay;
      barchart category=type response=mpg_highway / name="bar"
        stat=mean orient=horizontal
        colorbyfreq=true colorstat=pct;
      continuouslegend "bar" /
        title="Percent of Total Models Manufactured";
    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template=barchart;
run;
```
Example 5: Bar-Line Chart

This example uses the BARCHART and LINECHART statements to overlay a bar chart and line chart to create a bar-line chart. The following figure was generated by the “Example Program” on page 279.

Example Program

```sas
proc template;
    define statgraph barline;
    begingraph;
        entrytitle "Volume and Closing Price in 1H2001";
        layout overlay /
            xaxisopts=(label="Month" timeopts=(tickvalueformat=monname3.))
            yaxisopts=(label="Volume (Millions)" offsetmin=0 linearopts=(tickvaluesequence=(start=0 end=120 increment=20) tickvaluepriority=true))
            y2axisopts=(label="Close" offsetmin=0 linearopts=(viewmin=50 viewmax=75));
        barchart category=date response=eval(volume/1000000) / name="bar" legendlabel="Volume" datatransparency=0.6;
        linechart category=date response=close / legendlabel="Closing Price" vertexlabel=true vertexlabelattrs=(color=darkblue weight=bold) name="line" stat=sum yaxis=y2 display=(line markers) markerattrs=(symbol=circlefilled color=darkblue) lineattrs=(color=darkblue);
        discretelegend "bar" "line";
    endlayout;
endgraph;
end;
run;
```
BARCHARTPARM Statement

Creates a bar chart specified by pre-summarized data.

Requirement: The input data must be pre-summarized, with appropriate summarization statistics (sum, mean, and so on) computed for the RESPONSE column.

Tips: For charts that have a large number of bars that are very close together, slight variations in spacing that normally occur due to integer rounding can become more obvious. Subpixel rendering provides more precise bar spacing in that case. In SAS 9.4M2 and in earlier releases, specify SUBPIXEL=ON in the BEGINGRAPH statement to enable subpixel rendering. See SUBPIXEL= on page 35. Starting with SAS 9.4M3, subpixel rendering is enabled by default.

To disable subpixel rendering in SAS 9.4M3 and in later releases, specify SUBPIXEL=OFF in the BEINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

BARCHARTPARM CATEGORY=column | expression
RESPONSE=numeric-column | expression </options>;

Summary of Optional Arguments

Appearance options

BARWIDTH=number
specifies the width of a bar as a ratio of the maximum possible width.

BASELINEATTRS=style-element | (line-options)
specifies the appearance of the baseline.

CLUSTERWIDTH=number
specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the column or range attribute variable to use to map the bar colors to a continuous color gradient.

CONNECTATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the bar connect lines.

CONNECTBREAK=TRUE | FALSE

proc sgrender data=sashelp.stocks template=barline;
    where stock eq "Microsoft" and year(date) eq 2001 and 
    month(date) le 6;
    format date monname3. close dollar6.0;
run;
specifies whether the connect line is broken for values that have no observations.

**DATASKIN=** NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the filled bars.

**DATATRANSPARENCY=** number

specifies the degree of the transparency of the bar fill, bar outline, error bars, connect line, and data labels, if displayed.

**DISPLAY=(**display-options**) | STANDARD | ALL**

specifies which bar features to display.

**DISPLAYBASELINE=** ON | OFF | AUTO

specifies whether the baseline is displayed.

**DISPLAYZEROLENGTHBAR=** TRUE | FALSE

specifies whether zero-length bars are drawn.

**ERRORBARATTRS=** style-element | style-element (line-options) | (line-options)

specifies the attributes of the error bars that are associated with the data points.

**ERRORBARCAPSCALE=** positive-number

specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

**ERRORBARCAPSHAPE=** SERIF | NONE

specifies whether the error bars have a serif cap.

**ERRORLOWER=** numeric-column | expression

specifies the values of the lower endpoints on the Y error bars.

**ERRORUPPER=** numeric-column | expression

specifies the values of the upper endpoints on the Y error bars.

**FILLATTRS=** style-element | style-element (fill-options) | (fill-options)

specifies the appearance of the filled bar area.

**FILLENDCOLOR=** color

specifies the end color of the color gradient for the bar fill.

**FILLPATTERNATTRS=** style-element | (fill-pattern-options)

specifies the appearance of the pattern-filled areas.

**FILLTYPE=** SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT

specifies the bar fill type.

**INDEX=** positive-integer-column | expression

specifies indices for mapping bar attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**INTERVALBARWIDTH=** dimension

specifies the width of the bars in an interval bar chart as a ratio of the interval width.

**ORIENT=** VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.

**OUTLINEATTRS=** style-element | style-element (line-options) | (line-options)

specifies the appearance of the bar outlines.

**TARGET=** numeric-column | expression

specifies the target value for each bar.

### Axes options

**BASELINEINTERCEPT=** number

specifies the response axis intercept for the baseline.

**PRIMARY=** TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features.

\textbf{XAXIS=}X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

\textbf{YAXIS=}Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

\textbf{Data tip options}

\textbf{ROLENAME=} (role-name-list)
specifies user-defined roles that can be used to display information in the data tips.

\textbf{TIP=} (role-list) | NONE
specifies the information to display when the cursor is positioned over a bar.

\textbf{TIPFORMAT=} (role-format-list)
specifies display formats for tip columns.

\textbf{TIPLABEL=} (role-label-list)
specifies display labels for tip columns.

\textbf{Label options}

\textbf{DATALABEL=} column | expression
specifies the label that appears at the end of each bar.

\textbf{DATALABELATTRS=} style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the labels that are specified in the DATALABEL= option.

\textbf{DATALABELFITPOLICY=} AUTO | NONE | ROTATE | SPLIT | SPLITALWAYS | INSIDEPREFERRED
specifies a policy for avoiding collisions among the bar labels when labels are displayed.

\textbf{DATALABELSPLITCHAR=} "character-list"
specifies one or more characters on which the data labels can be split.

\textbf{DATALABELSPLITCHARDROP=} TRUE | FALSE
specifies whether the split characters are included in the displayed data labels.

\textbf{DATALABELTYPE=} AUTO | COLUMN
specifies whether the data labels display the RESPONSE values or the values of the column that is specified by the DATALABEL= option.

\textbf{LEGENDLABEL=} "string"
specifies a label to be used in a discrete legend for this plot.

\textbf{SEGMENTLABELATTRS=} style-element | style-element (text-options) | (text-options)
specifies the text properties of the text for the bar segment label.

\textbf{SEGMENTLABELFITPOLICY=} NONE | NOCLIP | THIN
specifies a policy for fitting the bar segment labels within the bar segments.

\textbf{SEGMENTLABELFORMAT=} format
specifies the text format for the bar segment labels.

\textbf{SEGMENTLABELTYPE=} NONE | AUTO
specifies whether a label is displayed inside each bar segment.
Midpoint options

**DISCRETEOFFSET=** *number*

specifies an amount to offset all bars from the category midpoints.

**GROUP=** *column | discrete-attr-var | expression*

creates a separate bar segment or bar for each unique group value in the specified column.

**GROUP100=** *NONE | MAGNITUDE | POSITIVE*

displays the response values, normalized to 100%.

**GROUPDISPLAY=** *STACK | CLUSTER*

specifies how to display grouped bars.

**GROUPORDER=** *DATA | REVERSEDATA | ASCENDING | DESCENDING*

specifies the ordering of the groups within a category.

**INCLUDEMISSINGGROUP=** *TRUE | FALSE*

specifies whether missing values of the group variable are included in the plot.

**ODS** options

**URL=** *string-column*

specifies an HTML page to display when the bar is selected.

**Plot reference options**

**NAME=** "*string"*

assigns a name to this plot statement for reference in other template statements.

**Required Arguments**

**CATEGORY=** *column | expression*

specifies the column for the unique category values. All values are treated as discrete.

**Note** You can use X= as an alternative to CATEGORY=. If you use X=, then be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options will recognize X as the category role and not CATEGORY in that case.

**RESPONSE=** *numeric-column | expression*

specifies the column for the response values.

**Note** You can use Y= as an alternative to RESPONSE=. If you use Y=, then be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options will recognize Y as the response role and not RESPONSE in that case.

**Optional Arguments**

**BARWIDTH=** *number*

specifies the width of a bar as a ratio of the maximum possible width.

**Default** 0.85

**Range** 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

**Interaction** Starting with SAS 9.4M3, the INTERVALBARWIDTH= option overrides this option for an interval bar chart.
Notes  This option is needed only to change the default behavior.

By default, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Tip  To remove any inter-bar gap, set BARWIDTH=1.

BASELINEATTRS=style-element | (line-options)
specifies the appearance of the baseline.

Notes  The baseline is always drawn by default.

When style-element is specified, only the style element’s COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip  To suppress the baseline, set the line thickness to 0:
baselineattrs=(thickness=0)

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

BASELINEINTERCEPT=number
specifies the response axis intercept for the baseline. Prior to SAS 9.4M5, the baseline is always displayed in the chart, whether for a specified value or for the default value. Starting with SAS 9.4M5, the baseline display is controlled by the DISPLAYBASELINE= option, which is ON by default. When the BASELINEINTERCEPT= option is used, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.

Default  0

Interactions  If GROUPDISPLAY=STACK is specified, then this option is ignored and the baseline is not displayed.

If necessary, the response axis data range is extended to include the baseline intercept. When a logarithmic response axis is requested and BASELINEINTERCEPT= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINEINTERCEPT= to a positive value.

When DISPLAYBASELINE=AUTO is in effect, the baseline is not displayed if the baseline intercept is equal to the minimum or maximum value of the response-axis range.
Note  Label positions are automatically adjusted to prevent the labels from overlapping.

Tips  Control the appearance of the baseline with the BASELINEATTRS= option.

To suppress the baseline prior to SAS 9.4M5, use the BASELINEATTRS= option to set the line thickness to 0. Starting with SAS 9.4M5, specify DISPLAYBASELINE=OFF.

The baseline does not add a tick or a tick value to the axis. To label the baseline, use a REFERENCeline statement to overlay a line with the same X or Y value and include the CURVELABEL= option to specify the label text.

**CLUSTERWIDTH=number**

specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

```
Cluster  Cluster  Cluster  Cluster  Cluster  Cluster  Cluster

| Type       |   |   |   |   |   |   |
|------------|---|---|---|---|---|---|---|
| SUV        | 20| 20| 20| 20| 20| 20| 20|
| Sedan      | 30| 30| 30| 30| 30| 30| 30|
| Sports     | 40| 40| 40| 40| 40| 40| 40|
| Wagon      | 50| 50| 50| 50| 50| 50| 50|
| Truck      | 60| 60| 60| 60| 60| 60| 60|
| Hybrid     | 70| 70| 70| 70| 70| 70| 70|

MPG (Highway)

Regions:  Asia  Europe  USA
```

Default  0.85

Range  0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Requirement  For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

Interaction  When GROUPDISPLAY=CLUSTER, the default BARWIDTH is 1.0.

**COLORMODEL=color-ramp-style-element | (color-list)**

specifies a color ramp to use with the COLORRESPONSE= option.

color-ramp-style-element

specifies the name of a color-ramp style element.

The style element should contain these style attributes:
STARTCOLOR
specifies the color for the smallest data value of the COLORRESPONSE= column.

NEUTRALCOLOR
specifies the color for the midpoint of the range of the COLORRESPONSE= column.

ENDCOLOR
specifies the color for the highest data value of the COLORRESPONSE= column.

(color-list)
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement  The list of colors must be enclosed in parentheses.

See  “color ” on page 1480

Defaults  For outline-only bars, the ThreeColorAltRamp style element
For bars with fill, the ThreeColorRamp style element

Interaction  For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip  Use the DISPLAY= option to specify whether outlines and fills are displayed.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the column or range attribute variable to use to map the bar colors to a continuous color gradient.

Note:  This feature applies to SAS 9.4M3 and to later releases.

range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“RANGEATTRVAR Statement”

Restriction  A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See  “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. Each bar is colored using one color from the gradient range. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Requirement  For a grouped plot, the COLORRESPONSE values should remain constant for each group value. If the COLORRESPONSE column
has multiple values for a single GROUP value, unexpected results might occur.

**Interactions**

When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

When fill, fill pattern, or both are displayed, this option overrides suboption COLOR= in the FILLATTRS= option and in the FILLPATTERNATTRS= option and varies the color according to the color gradient or the attribute map.

When only the outlines are displayed, this option overrides suboption COLOR= in the OUTLINEATTRS= option and varies the outline color according to the color gradient or the attribute map.

**Tips**

To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

Use the FILLTYPE= option to specify whether each bar is filled with a solid color or with a gradient color.

For a numeric column or expression, the ThreeColorRamp style element defines the fill color gradient, and the ThreeColorAltRamp style element defines the outline color gradient.

**CONNECTATTRS=**

<table>
<thead>
<tr>
<th>style-element</th>
<th>style-element (line-options)</th>
<th>(line-options)</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies the appearance of the bar connect lines.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default: The GraphConnectLine style element.

See "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Line Options" on page 1522 for available line-options.

**CONNECTBREAK=**

<table>
<thead>
<tr>
<th>TRUE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies whether the connect line is broken for values that have no observations.</td>
<td></td>
</tr>
</tbody>
</table>

Note: This feature applies to SAS 9.4M5 and to later releases.

Default: FALSE

Requirement: DISPLAY= must include CONNECT for this option to have any effect.

Interaction: This option is ignored when the GROUP= option is in effect.

**DATALABEL=**

<table>
<thead>
<tr>
<th>column</th>
<th>expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies the label that appears at the end of each bar.</td>
<td></td>
</tr>
</tbody>
</table>

Restriction: This option is ignored when one or both of the ERRORLOWER= and ERRORUPPER= options are in effect.

Interactions: Starting with SAS 9.4M2, this option is ignored when DATLABELTYPE=AUTO.
When the \texttt{GROUP=} option is in effect, the data label values are displayed only when \texttt{GROUPDISPLAY=}CLUSTER.

If the \texttt{GROUP=} option is in effect and there are multiple input observations per bar for the \texttt{GROUP=} column, then the value for the \texttt{DATALABEL=} column should be the same for each observation that is on the same bar.

\textbf{Tip} The data-label format is the format that is assigned to \texttt{column} or is derived from \texttt{BEST6} if no format is assigned. To change the data-label format, apply the new format to \texttt{column}.

\textbf{DATALABELATTRS=style-element | style-element (text-options) | (text-options)} specifies the color and font attributes of the labels that are specified in the \texttt{DATALABEL=} option.

\textbf{Default} The GraphDataText style element.

\textbf{Interaction} For this option to take effect, the \texttt{DATALABEL=} option must also be used.

\textbf{See} “General Syntax for Attribute Options” on page 1519 for the syntax on using a \texttt{style-element}.

“Text Options” on page 1525 for available \texttt{text-options}.

\textbf{DATALABELFITPOLICY=\texttt{AUTO} | \texttt{NONE} | \texttt{ROTATE} | \texttt{SPLIT} | \texttt{SPLITALWAYS} | \texttt{INSIDEPREFERRED}} specifies a policy for avoiding collisions among the bar labels when labels are displayed.

\texttt{AUTO} selects a collision avoidance policy based on the chart orientation and data type.

For a numeric column with \texttt{ORIENT=}VERTICAL, AUTO rotates the labels if they do not fit the midpoint spacing. For a character column, AUTO splits the labels if they do not fit the midpoint spacing.

\textbf{Note} When \texttt{ORIENT=}HORIZONTAL, AUTO always draws the labels horizontally.

\textbf{Tip} If character labels do not fit after splitting, then try using \texttt{ROTATE} instead of AUTO.

\textbf{See} \texttt{ORIENT=} on page 308 for information about chart orientation.

\texttt{BARWIDTH=} for information about bar spacing.

\texttt{NONE} does not attempt to fit bar labels that collide.

\texttt{ROTATE} rotates the bar labels for vertical bars if the labels collide in the available width.

\textbf{Requirement} The chart orientation must be vertical (\texttt{ORIENT=}VERTICAL).
SPLIT
splits the label for vertical bars at a split character only if a split is needed at that character in order to make the label fit the available space.

No split occurs at split characters that occur where a split is not needed. If the label does not contain any of the specified split characters, then a split does not occur. In that case, if the label does not fit the available space, then it might collide with the adjoining labels.

Requirement The chart orientation must be vertical (ORIENT=VERTICAL).

See the DATALABELSPLITCHAR= option for information about specifying the split characters.

SPLITALWAYS
splits the label for vertical bars at every occurrence of a split character.

If the label does not contain any of the specified split characters, then a split does not occur.

Requirement The chart orientation must be vertical (ORIENT=VERTICAL).

See the DATALABELSPLITCHAR= option for information about specifying the split characters.

INSIDE PREFERRED
attempts to place the data labels in a horizontal bar chart inside the bars.

Note: This feature applies to SAS 9.4M5 and to later releases.

Restrictions This option applies to non-grouped or cluster-grouped horizontal bar charts only.

Label splitting is not supported when a label is placed inside a bar.

Requirement For a grouped horizontal bar chart, the GROUPDISPLAY=CLUSTER option must be in effect. Otherwise, the INSIDE PREFERRED policy is ignored.

Interaction If SEGMENTLABEL=TRUE is in effect, the INSIDE PREFERRED policy is ignored, and the labels are placed outside.

Notes If any bar label does not fit inside its bar because the font height is too high, all of the bar labels are placed outside.

If any bar label cannot fit inside the length of its bar, that label is placed outside of its bar and is fit using the AUTO fit policy.

If the outside labels collide because the font height is greater than the midpoint space, then all labels are dropped.

Here is an example of a vertical bar chart where DATALABELFITPOLICY=AUTO and a numeric column is used as the data labels.
In this case, AUTO rotates the numeric labels to avoid collision.

In some cases, if one or more labels collide when the specified fit policy is used, then all of the labels are dropped from the display. When that occurs, the following warning message is written to the SAS log:

WARNING: The bar labels are suppressed. Use DATALABELFITPOLICY=NONE to force the labels to be displayed.

Here is an example of a horizontal bar chart where DATALABELFITPOLICY=INSIDEPREFERRED and a numeric column is used as the data labels.

Default:  AUTO

 Requirement: The DATALABEL= option must also be specified.
Interaction

When `DATALABELTYPE=AUTO` is in effect, for a vertical bar chart, only AUTO, NONE, and ROTATE are valid. All other values revert to AUTO. For a horizontal bar chart, only AUTO and NONE are valid. All other values revert to AUTO.

`DATALABELSPLITCHAR="character-list"`

specifies one or more characters on which the data labels can be split. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When `DATALABELFITPOLICY=SPLIT` and a data label collision is detected, the data label is split on a specified split character only if a split is needed at that point in order to make the label fit. In that case, a split might not occur on every split character. When `DATALABELFITPOLICY=SPLITALWAYS`, the data label is split unconditionally on every occurrence of a split character. If the data label does not contain any of the specified split characters, then the label is not split.

"character-list"

specifies one or more characters with no space between each character and enclosed in quotation marks.

Default

A blank space

Requirements

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```
datalabelsplitchar="abc"
```

The `DATALABELFITPOLICY=` option must specify SPLIT or SPLITALWAYS.

Interactions

The `DATALABELFITPOLICY=` option specifies the policy that is used to manage the split behavior of the data label.

The `DATALABELSPLITCHARDROP=` option specifies whether the split characters are included in the displayed data label or are dropped.

Notes

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

`DATALABELSPLITCHARDROP=TRUE | FALSE`

specifies whether the split characters are included in the displayed data labels.

TRUE

drops a split character from the display when a split occurs at that character.

Split characters at which a split does not occur are left in place. The `DATALABELFITPOLICY=` option determines where the labels are split. When `DATALABELFITPOLICY=SPLIT`, each label is split at a split character only where a split is needed in order to make the label fit the available space. At each split point, the split character is dropped, and the characters that follow
the split character, up to but not including the split character at the next split point, are wrapped to the following line.

When DATALABELFITPOLICY=SPLITALWAYS, each label is split at every instance of a split character. All of the split characters are dropped. The characters that follow each split character, up to but not including the next split character, are wrapped to the next line.

The following figure shows how label Product*Group*1 is split when the DATALABELSPLITCHARDROP=TRUE and DATALABELSPLITCHAR="*" options are specified with the SPLIT and SPLITALWAYS fit policies.

<table>
<thead>
<tr>
<th>DATALABELFITPOLICY=SPLIT</th>
<th>Product<em>Group</em>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATALABELFITPOLICY=SPLITALWAYS</td>
<td>Product<em>Group</em>1</td>
</tr>
</tbody>
</table>

In this example, when DATALABELFITPOLICY=SPLIT, the label is split at the first occurrence of the asterisk in order to make the label fit. No split is needed at the second asterisk. The first asterisk is dropped, and Group*1 wraps to the next line. Notice that the second asterisk is not dropped in this case. When DATALABELFITPOLICY=SPLITALWAYS, the label is split at every occurrence of the asterisk. In this case, both asterisks are dropped, and the characters that follow each asterisk wrap to the next line.

FALSE includes the split characters in the data label display.

The DATALABELFITPOLICY= option determines how the split characters are displayed. When DATALABELFITPOLICY=SPLIT, each data label is split at a split character only where a split is needed in order to make the label fit the available space. A split might not occur at every split character in the label. At each split point, the split character remains as the last character in the current line. The characters that follow the split character, up to and including the split character at the next split point, are then wrapped to the following line. This process repeats until the entire data label is displayed.

When DATALABELFITPOLICY=SPLITALWAYS, each data label is split at every instance of a split character in the label regardless of whether a split is actually needed. Each split character remains as the last character in the current line. The characters that follow each split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows how label Product*Group*1 is split when the DATALABELSPLITCHARDROP=FALSE and DATALABELSPLITCHAR="*" options are specified with the SPLIT and SPLITALWAYS fit policies.

<table>
<thead>
<tr>
<th>DATALABELFITPOLICY=SPLIT</th>
<th>Product<em>Group</em>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATALABELFITPOLICY=SPLITALWAYS</td>
<td>Product<em>Group</em>1</td>
</tr>
</tbody>
</table>

In this example, when DATALABELFITPOLICY=SPLIT, the label is split at the first occurrence of the asterisk in order to make the label fit. No split is
needed at the second asterisk. The characters that follow the first asterisk wrap to the next line. When DATALABELFITPOLICY=SPLITALWAYS, the label is split at every occurrence of the asterisk. Each asterisk remains as the last character in the current line, and the characters that follow are wrapped to the next line.

Default

**TRUE.** A split character is dropped from the data-label display when a split occurs at that character.

Requirements

The DATALABEL= option must also be specified.

The DATALABELFITPOLICY= option must specify SPLIT or SPLITALWAYS.

Interaction

The DATALABELSPLITCHAR= option specifies the split characters.

See

"boolean " on page 1479 for other Boolean values that you can use.

**DATALABELTYPE=** AUTO | COLUMN

specifies whether the data labels display the RESPONSE values or the values of the column that is specified by the DATALABEL= option.

**Note:** This feature applies to **SAS 9.4M2** and to later releases.

**AUTO**

the system determines how the labels are displayed.

The system displays the labels as follows:

- For an ungrouped bar chart, the summarized value for each bar is placed above the bar.
- For a grouped bar chart with stacked bars, the total of the summarized segment values for each bar is placed above the segmented bar.
- For a grouped bar chart with clustered bars, the summarized value for each bar in the cluster is placed above the bar.

**Interactions**

AUTO overrides the DATALABEL= option.

When AUTO is in effect, some data-label fit policies are unavailable. See DATALABELFITPOLICY=.

**COLUMN**

the data labels display the DATALABEL= column values.

**Interaction**

The DATALABEL= option must be specified for COLUMN to have any effect. If the DATALABEL= option is not specified, AUTO is used instead.

**Default**

**COLUMN**

**DATASKIN=** NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the filled bars. The following figure shows bars with each of the skins applied.
The `DATASKIN=` option value that is specified in the `BEGINGRAPH` statement. If not specified, then the `GraphSkins:DataSkin` style element value is used.

**Restriction**
Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the `DATASKINMAX=` option in your ODS GRAPHICS statement to increase the maximum limit.

**Requirement**
For this option to have any effect, the fill must be enabled by the ODS style or the `DISPLAY=` option.

**Interactions**
This option overrides the `BEGINGRAPH` statement `DATASKIN=` option.

The data skin appearance is based on the `FILLATTRS=` color.

When a data skin is applied, all bar outlines are set by the skin, and the `OUTLINEATTRS=` option is ignored.

When `FILLTYPE=GRADIENT` is in effect, `DATASKIN=SHEEN` is ignored. In that case, use one of the other skins.

**DATATRANSPARENCY=number**

specifies the degree of the transparency of the bar fill, bar outline, error bars, connect line, and data labels, if displayed.

**Default**
0

**Range**
0–1, where 0 is opaque and 1 is entirely transparent

**Tip**
The `FILLATTRS=` option can be used to set transparency for just the bar fills. You can combine this option with `FILLATTRS=` to set one transparency for the bar outlines, error bars, and connect lines but a different transparency for the bar fills. Example:

```
data transparency=0.2 fillattrs=(transparency=0.6)
```

**DISCRETEOFFSET=number**

specifies an amount to offset all bars from the category midpoints.
**DISPLAY=(display-options) | STANDARD | ALL**

specifies which bar features to display.

*(display-options)*

a space-separated list of one or more of the following options enclosed in parentheses:

- **CONNECT**
  - displays the connect lines, which are line segments that connect adjacent midpoints at the end of each bar

- **FILL**
  - displays the bar fill color

- **FILLPATTERN**
  - displays the bar fill pattern

  **Tip** For this option to have any effect, the active ODS style must specify a fill pattern or a fill pattern must be specified with the **FILLPATTERNATTRS=** option.

- **OUTLINE**
  - displays the bar outline

**STANDARD**

specifies FILL and OUTLINE

**ALL**

specifies all features: CONNECT, FILL, FILLPATTERN, and OUTLINE

**Default** The GraphBar:DisplayOpts style reference.

**Interaction** Connect lines are not drawn when the **GROUP=** option is in effect.

**Note** Error bars are automatically displayed whenever the **ERRORUPPER=** or **ERRORLOWER=** options are specified.

**Tips** Use the **OUTLINEATTRS=** , **FILLATTRS=** , and **FILLPATTERNATTRS=** options to control the appearance of the
bars. Use `CONNECTATTRS=` to control the appearance of the connect lines.

Both `FILL` and `FILLPATTERN` can be specified to combine solid fills and pattern fills in the bars.

**DISPLAYBASELINE=** `ON` | `OFF` | `AUTO`

specifies whether the baseline is displayed.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

- **ON** always displays the baseline.
- **OFF** does not display the baseline.
- **AUTO** displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

This is the typical case when the bar chart includes both positive and negative bars, and the default baseline intercept of 0 is used. If the baseline intercept is equal to the minimum or maximum value of the response-axis range, the baseline is not displayed.

**Default** ON

**Tip** By default, the baseline intercept is 0. Use the `BASELINEINTERCEPT=` option to change the baseline intercept.

**DISPLAYZEROLENGTHBAR=** `TRUE` | `FALSE`

specifies whether zero-length bars are drawn.

**Note:** This feature applies to SAS 9.4M3 and to later releases.

A zero-length bar is displayed as a line spanning the normal bar width at the bar-chart baseline on the response axis. When this option is set to `TRUE`, zero-length bars are displayed. Otherwise, they are suppressed. The following figure shows a simple example of each outcome. In the figure, the plot wall outline, category axis line, and bar-chart baseline are suppressed for clarity.

**Default** `TRUE`

**Interaction** This option is ignored when the `GROUP=` and `GROUPDISPLAY=STACK` options are in effect. In that case, zero-length bar segments are drawn.
Note: When this option is set to FALSE, the bar is not drawn, but other elements associated with the bar such as the target bar, the error bar, the bar label, and the data label, are drawn.

Tip: This option is useful when the bar-chart baseline is suppressed.

**ERRORBARATTRS=**<em>style-element | style-element (line-options) | (line-options)</em>
specifies the attributes of the error bars that are associated with the data points.

**Default**
The GraphError style element contrastColor, lineStyle, and lineThickness attributes.

**Interaction**
For this option to take effect, error bars must be displayed by the ERRORLOWER=, ERRORUPPER= options.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**ERRORBARCAPSCALE=**<em>positive-number</em>
specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Default**
1

**Interaction**
This option is ignored when ERRORBARCAPSHAPE=NONE is in effect.

**Tips**
Use the ERRORBARCAPSHAPE= option to specify the cap shape.

The thickness (height) of the SERIF, LINE, and BRACKET is determined by the thickness specified in the ERRORBARATTRS= option. The cap line pattern is always solid.

**Examples**
Double the size of the error-bar caps:
errorbarcapscale=2

Reduce the size of the error-bar caps by 50%:
errorbarcapscale=0.5

**ERRORBARCAPSHAPE=**<em>SERIF | NONE</em>
specifies whether the error bars have a serif cap.

<table>
<thead>
<tr>
<th>SERIF</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Default**
The GraphError:CapStyle style reference. If attribute CapStyle is not defined in the active style, then SERIF is the default value.
Interaction  Starting with SAS 9.4M5, when ERRORBARCAPSCALE= is in effect, the display width for SERIF caps is the cap's default width multiplied by the value specified in ERRORBARCAPSCALE=.

Tip  The appearance of the error bars is controlled by the ERRORBARATTRS= option.

**ERRORLOWER=**`numeric-column | expression`  
specifies the values of the lower endpoints on the Y error bars.

<table>
<thead>
<tr>
<th>Default</th>
<th>The lower segment of the error bars is not drawn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>The error bar values must be absolute data values, not data values relative to the value of the bar.</td>
</tr>
<tr>
<td>Interaction</td>
<td>If the GROUP= option is specified with GROUPDISPLAY=STACK or with GROUP100=POSITIVE or MAGNITUDE, then this option is ignored.</td>
</tr>
<tr>
<td>Tip</td>
<td>You can use the ERRORBARATTRS= option to control the appearance of the error bars.</td>
</tr>
</tbody>
</table>

**ERRORUPPER=**`numeric-column | expression`  
specifies the values of the upper endpoints on the Y error bars.

<table>
<thead>
<tr>
<th>Default</th>
<th>The upper segment of the error bars is not drawn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>The error bar values must be absolute data values, not data values relative to the value of the bar.</td>
</tr>
<tr>
<td>Interaction</td>
<td>If the GROUP= option is specified with GROUPDISPLAY=STACK or with GROUP100=POSITIVE or MAGNITUDE, then this option is ignored.</td>
</tr>
<tr>
<td>Tip</td>
<td>You can use the ERRORBARATTRS= option to control the appearance of the error bars.</td>
</tr>
</tbody>
</table>

**FILLATTRS=**`style-element | style-element (fill-options) | (fill-options)`  
specifies the appearance of the filled bar area.

<table>
<thead>
<tr>
<th>Defaults</th>
<th>For non-grouped data, the GraphDataDefault:Color style reference. For grouped data, the GraphData1:Color–GraphDataN:Color style references.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>When COLORRESPONSE= is in effect and the DISPLAY= option enables FILL display, the FILLATTRS= suboption COLOR= is ignored, and the bar fill colors vary according to the gradient.</td>
</tr>
<tr>
<td>Tip</td>
<td>The DATATRANSPARENCY= option sets the transparency for bar fills, bar outlines, error bars, and connect lines. You can combine this option with DATATRANSPARENCY= to set one transparency for the bar outlines, error bars, and connect lines but a different transparency for the bar fills. Example: datatransparency=0.2 fillattrs=(transparency=0.6)</td>
</tr>
</tbody>
</table>
See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

FILLENDCOLOR=color
specifies the end color of the color gradient for the bar fill.

Note: This feature applies to SAS 9.4M5 and to later releases.

Default the graph wall background color, even when the wall is not displayed

Requirement FILLTYPE=COLORGRADIENT must be in effect. Otherwise, this option is ignored.

See “color” on page 1480

FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

Defaults For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

Restriction In SAS 9.4M4 and in earlier releases, the only styles that are delivered by SAS that support fill patterns are JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER. If style-element is specified and the active ODS style does not support fill patterns, this option is ignored.

Interactions For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

When COLORRESPONSE= is in effect and the DISPLAY= option enables FILLPATTERN display, the FILLPATTERNATTRS= suboption COLOR= is ignored, and the fill-pattern colors vary according to the gradient.

Tip In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Pattern Options” on page 1521 for available fill-pattern-options.

FILLTYPE=SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT
specifies the bar fill type.
Note: This feature applies to SAS 9.4M2 and to later releases.

**SOLID**
fills each bar with the color that is assigned to that bar.

**ALPHAGRADIENT**
fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.

Note: This feature applies to SAS 9.4M5 and to later releases.

**COLORGRADIENT**
fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the FILLENDCOLOR= option.

Note: This feature applies to SAS 9.4M5 and to later releases.

**Tips**
The fill color is determined by a style element or by the FILLATTRS= option COLOR= suboption. The initial transparency is determined by the DATATRANSPARENCY= option or by the FILLATTRS= option TRANSPARENCY= suboption.

For grouped plots, use the FILLATTRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**Alias**
GRADIENT

**Interaction**
The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

**Tips**
Use the DATATRANSPARENCY= option or the FILLATTRS= option TRANSPARENCY= suboption to set the initial transparency in the gradients.

For grouped plots, use the FILLATTRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**See**
"FILLENDCOLOR=color" on page 299

**GRADIENT**
fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.

Note: This option applies to SAS 9.4M2 through SAS 9.4M4. Starting with SAS 9.4M5, ALPHAGRADIENT replaces GRADIENT, and GRADIENT is changed to an alias of COLORGRADIENT.
The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

In SAS 9.4M2, FILLTYPE=GRADIENT is ignored when GROUPDISPLAY=STACK is in effect. Starting with SAS 9.4M3, FILLTYPE=COLORGRADIENT is honored in that case.

The initial fill color is determined by a style element or by the FILLATTRS= option COLOR= suboption. The initial transparency is determined by the DATATRANSPARENCY= option or by the FILLATTRS= option TRANSPARENCY= suboption.

For grouped plots, use the FILLATTRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

The DISPLAY= option must include FILL for this option to have any effect.

The output for SAS programs written prior to SAS 9.4M5 that specify FILLTYPE=GRADIENT might change starting with SAS 9.4M5. To restore the original appearance, specify FILLTYPE=ALPHAGRADIENT.

GROUP=column | discrete-atr-var | expression
creates a separate bar segment or bar for each unique group value in the specified column.

discrete-atr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

The bar display depends on the setting for the GROUPDISPLAY= option. For example, for a vertical bar chart with GROUPDISPLAY=STACK, bar segments are stacked to form the bar. The height of each segment represents the corresponding group value’s proportional contribution to the response value.

A distinct bar or bar segment is created for each group value by varying the visual attributes of the bar display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.
### Style Attributes That Control Default Group Appearance

<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern. If the active ODS style does not specify FillPattern, fill patterns are not displayed by default.</td>
</tr>
</tbody>
</table>

### Outline
- if the outline is enabled with fill color, fill pattern, or both, ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style
- if the outline is the only display feature enabled, ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element

1 The default display features are determined by the DisplayOpts attribute of the GraphBar style element.

### Interactions
Connect lines are not drawn for grouped data.

By default, the group values are mapped in the order of the data. Use the GROUPORDER= option to control the sorting order of the group values.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

When both the GROUP= and COLORRESPONSE= options are specified, the color attributes are controlled by the COLORRESPONSE= option.

### Note
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

### Tips
The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the bar outlines, but you could use the PATTERN= setting on the OUTLINEATTRS= option to assign the same line pattern to all bar outlines and connect lines.

Use the INDEX= option to alter the default sequence of colors, fill patterns, and line patterns.
GROUP100=NONE | MAGNITUDE | POSITIVE displays the response values, normalized to 100%.

Note: This feature applies to SAS 9.4M3 and to later releases.

NONE displays the data as received.

MAGNITUDE normalizes both the negative and positive values to 100% by magnitude, and displays the group values, preserving the sign.

The positive values are displayed above the bars for a vertical bar chart and on the right end for a horizontal bar chart. The negative values are displayed enclosed in parentheses below the bars for a vertical bar chart and on the left end for a horizontal bar chart.

The following figure illustrates the effect of MAGNITUDE on stacked bars in a vertical bar chart.

POSITIVE drops the negative values and normalizes only the positive values to 100%.

The following figure demonstrates the effect of POSITIVE on clustered bars in a vertical bar chart. This chart uses the same data as the chart in the previous figure.
Notice that the negative values are dropped from the chart.

**Default**

NONE

**Requirement**

The GROUP= option must be specified for this option to have any effect.

**Interaction**

Error bars are not drawn when GROUP100=POSITIVE or MAGNITUDE. See ERRORLOWER= and ERRORUPPER=.

**Note**

You can use this option with any value for the GROUPDISPLAY= option.

**Tip**

To display the values, specify DATALABELTYPE=AUTO.

**GROUPDISPLAY=STACK | CLUSTER**

specifies how to display grouped bars.

**STACK**

displays group values as stacked segments within the category bar.

![Average Mileage by Vehicle Type](image)

**CLUSTER**

displays group values as separate adjacent bars that replace the single category bar.

Each cluster of group values is centered at the category midpoint on the axis. This example illustrates the clusters and also how groups are displayed when they have an unequal number of unique values.
When you use the `DATALABEL=` option and the `GROUP=` option, the DATALABEL values are displayed for each bar when `GROUPDISPLAY=CLUSTER`. When `GROUPDISPLAY=STACK`, the whole bar is labeled at the top.

Error bars are not drawn when `GROUPDISPLAY=STACK`.

When the `TARGET=` and `GROUP=` options are in effect, the target values are not displayed when `GROUPDISPLAY=STACK`. In that case, you must specify `GROUPDISPLAY=CLUSTER` to display the target values.

When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

**GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING**

specifies the ordering of the groups within a category.

**DATA**
orders the groups within a category in the group-column data order.

**REVERSEDATA**
orders the groups within a category in the reverse group-column data order.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**Tip** This option is useful when you want to reverse the category axis.

**ASCENDING**
orders the groups within a category in ascending order.

**DESCENDING**
orders the groups within a category in descending order.

**Defaults** DATA if the data is a SAS data set.
ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

### Interactions

This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

### Notes

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

### INCLUDEMISSINGGROUP=TRUE | FALSE

specifies whether missing values of the group variable are included in the plot.

**Default**

TRUE

**Interaction**

For this option to take effect, the GROUP= option must also be specified.

**Tip**

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

### INDEX=positive-integer-column | expression

specifies indices for mapping bar attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**Requirements**

The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.
The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Notes**
The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip**
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 198.

**INTERVALBARWIDTH=** *dimension*

specifies the width of the bars in an interval bar chart as a ratio of the interval width.

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**Default**
The width specified by the BARWIDTH= option.

**Restriction**
This option applies only to an interval category axis. When the category axis is discrete, this option is ignored.

**Interaction**
When the category data is interval, this option overrides the BARWIDTH= option.

**Tips**
To make the category axis type linear or time, include TYPE=LINEAR or TYPE=TIME in the category axis options or assign the role of primary plot to a plot that makes the category axis linear or time.

The bar width that you specify with this option is honored even if the bars overlap. If the bars overlap, reduce the interval bar width or use the BARWIDTH= option instead.

**See**
“dimension" on page 1480

**LEGENDLABEL=** "*string*"

specifies a label to be used in a discrete legend for this plot.

**Default**
The response-variable label. If a label is not defined, then the response-variable name is used.

**Restriction**
This option applies only to an associated DISCRETELEGEND statement.
Interaction If the GROUP= option is specified, then this option is ignored.

NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction The string is used as the default legend label if the LEGENDLABEL= option is not used.

ORIENT=VERTICAL | HORIZONTAL
specifies the orientation of the Y axis and the bars.

Default VERTICAL

Notes When this option is set to VERTICAL, the category variable appears on the X (or X2) axis and the response variable appears on the Y (or Y2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

When this option is set to HORIZONTAL, the category variable appears on the Y (or Y2) axis and the response variable appears on the X (or X2) axis. To set the axis properties for this chart, you should use the appropriate axis options of the layout container.

If you change the orientation of the bar chart, then you should adjust the layout container’s axis options appropriately.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the bar outlines.

Defaults For non-grouped data, the ContrastColor, LineThickness, and LineStyle attributes of the GraphOutlines style element.

For grouped data and filled bars, the ContrastColor attribute of the GraphData1–GraphDataN style elements, and the LineThickness and LineStyle attributes of the GraphOutlines style element.

For grouped data and unfilled bars, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the LineThickness attribute of the GraphOutlines style element.

Interactions For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

If the DATASKIN= option applies a data skin, then this option is ignored.

When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the bar outline colors vary according to the gradient.
See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**
specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

(role-name-list)
a space-separated list of role-name = column pairs.

Example The following example assigns the column Obs to the user-defined role TIP:

ROLENAME=(TIP1=OBS)

Default No user-defined roles

Requirement The role names that you choose must be unique and different from the predefined roles CATEGORY or X, RESPONSE or Y, ERRORLOWER, ERRORUPPER, GROUP, and INDEX.

**SEGMENTLABELATTRS=**

specifies the text properties of the text for the bar segment label.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

Default The GraphDataText style element.

Interaction This option is ignored when SEGMENTLABELTYPE=NONE.

See “General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element.
**SEGMENTLABELFITPOLICY=NONE | NOCLIP | THIN**  
specifies a policy for fitting the bar segment labels within the bar segments.

Note: This feature applies to **SAS 9.4M2** and to later releases.

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NONE</strong></td>
<td>no attempt is made to fit each segment label within its bar. Long bar segment labels might overlap other graphical elements. The segment labels are not considered when the axis ranges are computed. As a result, segment labels that extend beyond the plot area are clipped.</td>
<td></td>
</tr>
<tr>
<td><strong>NOCLIP</strong></td>
<td>does not clip bar segment labels that extend beyond the plot area. Labels that do not fit within the plot area extend into the graph axis area and might overlap axis elements.</td>
<td></td>
</tr>
<tr>
<td><strong>THIN</strong></td>
<td>drops any bar segment label that does not fit within its segment. For a vertical bar chart, the label width must not exceed the bar width, and the text height must not exceed the segment height. For a horizontal bar chart, the label text height must not exceed the bar width, and the label length must not exceed the segment length.</td>
<td></td>
</tr>
</tbody>
</table>

Default: THIN

Interaction: This option is ignored when **SEGMENTLABELTYPE=NONE**.

**SEGMENTLABELFORMAT=**<format>  
specifies the text format for the bar segment labels.

Note: This feature applies to **SAS 9.4M2** and to later releases.

Default: If **DATALABELTYPE=AUTO** is in effect, the segment-label format is derived from the response column format. If **DATALABELTYPE=COLUMN** is in effect, the segment-label format is derived from the **DATALABEL=** column format. If no format is applied to the column, the label format is derived from **BEST6**. The derived format retains the precision of the column format and, if necessary, increases the format width to accommodate the column values.

Interaction: This option is ignored when **SEGMENTLABELTYPE=NONE**.

Note: When a segment-label format is specified with this option, the segment labels are formatted as specified by **format**. The specified format is not automatically expanded to accommodate values that are too wide.

Tip: If you want the segment-label format to expand automatically for the column values, specify the format for the response column or **DATALABEL=** column rather than in this option.
SEGMENTLABELTYPE=NONE | AUTO

specifies whether a label is displayed inside each bar segment.

Note: This feature applies to SAS 9.4M2 and to later releases.

For an ungrouped bar chart or for a grouped bar chart with GROUPDISPLAY=CLUSTER, AUTO displays a bar label inside each bar. The label for each bar displays the value for that bar. For a grouped bar chart with GROUPDISPLAY=STACK, AUTO displays a label inside each bar segment. The label for each bar segment displays the value for that segment.

When this value is set to NONE, no labels are displayed inside the bars.

Default      NONE

Interaction The DATALABELTYPE= option determines whether the segment labels display the RESPONSE column values or the values of the column that is specified by the DATALABEL= option.

Note By default, the segment-label format is derived from the format that is assigned to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response column format and, if necessary, increases the format width to accommodate the column values.

Tips Use the SEGMENTLABELATTRS= option to modify the appearance of the label text.

Use the SEGMENTLABELFITPOLICY= option to specify a policy for fitting the labels inside the bars.

Use the SEGMENTLABELFORMAT= option to modify the format of the segment labels.

See “boolean ” on page 1479 for other Boolean values that you can use.

TARGET=numeric-column | expression

specifies the target value for each bar.
layout overlay;
  barchart category=type response=mpg_highway / barwidth=.8
target=mpg_city group=origin groupdisplay=cluster
  name='bar';
discretelegend 'bar';
endlayout;

The visual representation is a triangle with a line at the target value.

layout overlay;
  barchartparm category=type response=mpg_highway / barwidth=.8
target=mpg_city group=origin groupdisplay=cluster
  name='bar';
discretelegend 'bar';
endlayout;

The target color is that of the bar outline.

TIP=\(\text{role-list}\) | NONE

specifies the information to display when the cursor is positioned over a bar. If this option is used, then the information specified replaces all of the information that is displayed by default. You can specify roles for columns that do not contribute to the bar chart along with roles that do.

\(\text{role-list}\)

an ordered, space-separated list of unique BARCHARTParm roles and user-defined roles.

BARCHARTParm roles include CATEGORY or X, RESPONSE or Y, COLORRESPONSE, ERRORUPPER, ERRORLOWER, INDEX, GROUP, and DATALABEL.

Notes For the category and response roles, the TIP= option recognizes only the category and response arguments that you use in the BARCHARTParm statement. If you use the CATEGORY= and
RESPONSE= arguments, then you must specify roles CATEGORY and RESPONSE. Conversely, if you use the X= and Y= arguments, then you must specify roles X and Y.

The COLORRESPONSE role is valid starting with SAS 9.4M3.

Tip Use the ROLENAME= option to define user-defined roles.

NONE suppresses data tips and URLs (if requested) from the plot.

Default The columns assigned to these roles are automatically included in the data tip information: CATEGORY or X, RESPONSE or Y, COLORRESPONSE, ERRORLOWER, ERRORUPPER, and GROUP.

Requirement To generate data tips in the output, you must include an ODS GRAPHICS ON statement with the IMAGEMAP option specified, and you must write the output to the ODS HTML destination.

Interaction This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

Example The following example displays data tips for the columns assigned to the roles CATEGORY and RESPONSE as well as for the column Pct. The Pct column is not assigned to any pre-defined bar chart role, so it must first be assigned a role.

ROLENAME=(TIP1=PCT)
TIP=(TIP1 CATEGORY RESPONSE)

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction Only the roles that appear in the TIP= option are used.

Requirement A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.
role-label-list

a space-separated list of rolename = "string" pairs.

Example

```plaintext
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

Default
The column label or column name of the column assigned to the role.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

URL=string-column

specifies an HTML page to display when the bar is selected.

string-column

specifies a column that contains a valid HTML page reference (HREF) for each bar that is to have an active link.

Example

```
http://www.sas.com
```

Requirements
To generate selectable bars, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

If you are not using ODS Interactive HTML Graphics, you must place this HTML source code in an HTML file that is referenced by the URL option.

For non-grouped data, the values of the column are expected to be the same for each unique category value. If they are not, then only the first URL value for a given category value is used.

For grouped data, the values of the column are expected to be the same for each unique category and GROUP combination.

Interactions
This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tips
The URL value can be blank for some category values, meaning that no action is taken when the bars for those category values are selected.

The URL value can be the same for different category values, meaning that the same action is taken when the bars for those category values are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.
**XAXIS=X | X2**  
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.  
**Default**  
X  
**Interaction**  
The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

**YAXIS=Y | Y2**  
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.  
**Default**  
Y  
**Interaction**  
The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

---

**Details**

**Statement Description**

The input data for the BARCHARTPARM statement is expected to be pre-summarized, with appropriate summarization statistics (sum, mean, and so on) computed for the response column. When the category values are not unique, the display is not meaningful. Prior to **SAS 9.4M3**, the category axis must be discrete. Starting with **SAS 9.4M3**, the category axis can be discrete, linear, or time. The response axis in all cases is interval.

When the chart is oriented vertically, the X (or X2) axis is used for category and the Y (or Y2) axis is used for response. When the chart is oriented horizontally, the X (or X2) axis is used for response and the Y (or Y2) axis is used for category. (See **ORIENT=** on page 308.) If the chart is the primary chart, then any charts or plots that are overlaid with it must have similar axis types.

By default, if the CATEGORY= column is character, then the bars in the chart appear in the order in which the category values are present in the input data. If the CATEGORY= column is numeric, then the values are presented in ascending order.

**About the DISCRETEOFFSET= Option**

The **DISCRETEOFFSET=** option is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BARCHART statements are used with different response variables, then the bars for matching category values are centered on the midpoints and the bars are superimposed. To make it easier to distinguish among superimposed bars, you can assign a different **BARWIDTH=** setting to each BARCHARTPARM statement in the overlay:

```
layout overlay / cycleattrs=true
  xaxisopts=(display=(tickvalues))
  yaxisopts=(label="Revenue" offsetmax=0.2);

barchartparm category=year response=A_revenue / name="A"
  legendlabel="A" barwidth=0.8;

barchartparm category=year response=B_revenue / name="B"
```
legendlabel="B" barwidth=0.6;
barchartparm category=year response=C_revenue / name="C"
  legendlabel="C" barwidth=0.4;

discretelegend "A" "B" "C" / title="Product:"
  location=inside halign=right valign=top;
endlayout;

To place the different response values side by side, you can assign a different offset to each BARCHARTPARM statement. The BARWIDTH= option can be used with DISCRETEOFFSET= to create narrower bars that require less width within the plot area:

layout overlay / cycleattrs=true
  xaxisopts=(display=(tickvalues))
  yaxisopts=(label="Revenue" offsetmax=0.2);

barchartparm category=year response=A_revenue / name="A"
  legendlabel="A"
  discreteoffset=-0.3 barwidth=0.3;
barchartparm category=year response=B_revenue / name="B"
  legendlabel="B"
  discreteoffset=0   barwidth=0.3;
barchartparm category=year response=C_revenue / name="C"
  legendlabel="C"
  discreteoffset=+0.3 barwidth=0.3;

discretelegend "A" "B" "C" / title="Product:"
  location=inside halign=right valign=top;
endlayout;
Different combinations of DISCRETEOFFSET and BARWIDTH can be used to get the effect that you want. Gaps can be created between bars by providing a narrower bar width. Or, bars can be overlapped if the bar widths are increased in proportion to the discrete offset.

```
layout overlay / cycleattrs=true
   xaxisopts=(display=(tickvalues))
   yaxisopts=(label="Revenue" offsetmax=0.2);

barchartparm category=year response=A_revenue / name="A"
   legendlabel="A" datatransparency=0.2
   discreteoffset=-0.2 barwidth=0.5 ;

barchartparm category=year response=B_revenue / name="B"
   legendlabel="B" datatransparency=0.2
   discreteoffset=0  barwidth=0.5 ;

barchartparm category=year response=C_revenue / name="C"
   legendlabel="C" datatransparency=0.2
   discreteoffset=+0.2 barwidth=0.5 ;

discretelegend "A" "B" "C" / title="Product:
   location=inside halign=right valign=top;
endlayout;
```
Example: BARCHARTPARM Statement

The following graph was generated by the “Example Program” on page 318:

![Average Mileage by Vehicle Type](image)

Errorbars show +/- 1 Standard Error

Example Program

```plaintext
proc template;
  define statgraph barchartparm;
  begingraph;
    entrytitle "Average Mileage by Vehicle Type";
    entryfootnote halign=left
      "Error bars show +/- 1 Standard Error";
    layout overlay;
      barchartparm category=type response=mean /
        errorlower=eval(mean-stderr)
        errorupper=eval(mean+stderr) ;
    endlayout;
  endgraph;
end;
run;

/* create summarized data for barchartparm */
proc summary data=sashelp.cars nway;
  class type;
  var mpg_highway;
  output out=mileage mean=mean stderr=stderr ;
run;

proc sgrender data=mileage template=barchartparm;
run;
```
BIHISTOGRAM3DPARM Statement

Creates a three-dimensional bivariate histogram of three variables X, Y, and Z, where the values of X and Y have been gridded. The Z variable represents a response value for the frequency, percentage counts, or densities of each bin combination.

Restriction:
BIHISTOGRAM3DPARM does not support the data tips that are enabled by the IMAGEMAP= option in the ODS GRAPHICS statement.

Requirements:
The input data must be binned by both X and Y. That is, the values for X column and Y column must form a complete rectangular grid of bins. Input data with non-binned columns should be preprocessed with the KDE procedure (SAS/STAT), which enables you to set the number of bins for X and Y, or with a technique similar to that used in “Example: BIHISTOGRAM3DPARM Statement” on page 323.

The BIHISTOGRAM3DPARM statement must be specified within a LAYOUT OVERLAY3D statement and cannot be nested under an OVERLAY, OVERLAYEREQUATED, or PROTOTYPE layout.

The input data for Z= column must be nonnegative.

Notes:
In the plot display, the direction of the Z axis is upward rather than outward.
If the plot data is a CAS in-memory table, the data is sorted automatically by X and Y before it is plotted.

Syntax

BIHISTOGRAM3DPARM X=numeric-column | expression
Y=numeric-column | expression
Z=non-negative-numeric-column | expression <options>;

Summary of Optional Arguments

Appearance options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATATRANSPARENCY=number</td>
<td>specifies the degree of the transparency of the bins.</td>
</tr>
<tr>
<td>DISPLAY=(display-options)</td>
<td>STANDARD</td>
</tr>
<tr>
<td>FILLATTRS=style-element</td>
<td>style-element (fill-options)</td>
</tr>
<tr>
<td>OUTLINEATTRS=style-element</td>
<td>style-element (line-options)</td>
</tr>
<tr>
<td>XVALUES=MIDPOINTS</td>
<td>LEFTPOINTS</td>
</tr>
<tr>
<td>YVALUES=MIDPOINTS</td>
<td>LEFTPOINTS</td>
</tr>
</tbody>
</table>

Axes options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BINAXIS=TRUE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>
specifies whether to use bins as the basis for the axis tick marks.

**PRIMARY=TRUE | FALSE**

specifies that the data columns for this plot and the plot type be used for determining default axis features.

**Label options**

**ENDLABELS=TRUE | FALSE**

specifies whether the axis ticks and value labels are drawn at the bin endpoints (TRUE) or at the bin midpoints (FALSE).

**LEGENDLABEL="string"**

specifies a label to be used in a discrete legend for this plot.

**Plot reference options**

**NAME="string"**

assigns a name to this plot statement for reference in other template statements.

**Required Arguments**

**X=numeric-column | expression**

specifies the bin location of the numeric X values.

**Y=numeric-column | expression**

specifies the bin location of the numeric Y values.

**Z=nonnegative-numeric-column | expression**

specifies the response values, such as the frequency counts, percentages, or densities.

**Optional Arguments**

**BINAXIS=TRUE | FALSE**

specifies whether to use bins as the basis for the axis tick marks.

**TRUE**

specifies that the ENDLABELS= option determines how the axis ticks and value labels are displayed.

**FALSE**

specifies that standard axes are used. Bin boundaries and midpoints that are set by the ENDLABELS= option are ignored.

**Default**

**TRUE**

**Interactions**

For this option to take effect, this plot must be the primary plot in the parent OVERLAY3D layout. For more information, see the **PRIMARY= option**.

When this option is set to TRUE, some X-axis options that are set on the parent layout might not apply, such as INTEGER=, TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

**See**

“**ENDLABELS=TRUE | FALSE**” on page 321

“boolean “ on page 1479 for other Boolean values that you can use.
DATATRANSPARENCY=number
specifies the degree of the transparency of the bins.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

DISPLAY=(display-options) | STANDARD | ALL
specifies which bin features to display.

(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

FILL
displays the bin fill color

OUTLINE
displays the bin outline

STANDARD
specifies FILL

ALL
specifies FILL and OUTLINE

Default STANDARD

Tip Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bins.

ENDLABELS=TRUE | FALSE
specifies whether the axis ticks and value labels are drawn at the bin endpoints (TRUE) or at the bin midpoints (FALSE).

Default FALSE.

Interactions This option is ignored if this plot is not the primary plot in the parent layout. For more information, see the PRIMARY= option.

This option is ignored if BINAXIS= FALSE. By default, BINAXIS=TRUE.

See “boolean “ on page 1479 for other Boolean values that you can use.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the filled bins.

Default The GraphDataDefault style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

LEGENDLABEL=“string”
specifies a label to be used in a discrete legend for this plot.

Default The string specified on the NAME= option.
Restriction
This option applies only to an associated `DISCRETELEGEND` statement.

**NAME=“string”**
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**
The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**
The string is used as the default legend label if the `LEGENDLABEL=` option is not used.

**OUTLINEATTRS=**
`style-element | style-element (line-options) | (line-options)`
specifies the appearance of the bin outlines.

**Default**
The GraphOutlines style element.

**Restriction**
This option uses only the color specification in the style element or line options. The line pattern and line thickness specifications are ignored.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`

**COLOR=** on page 1522

**PRIMARY=**
`TRUE | FALSE`
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**
FALSE

**Restriction**
This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**See**
“When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**XVALUES=**
`MIDPOINTS | LEFTPOINTS | RIGHTPOINTS`
specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

**Default**
MIDPOINTS

**YVALUES=**
`MIDPOINTS | LEFTPOINTS | RIGHTPOINTS`
specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

**Default**
MIDPOINTS
Details

Because the BIHISTOGRAM3DPARM statement does not perform a binning computation on the input columns, you must pre-bin the data. In addition, input data for the statement must be binned by both X and Y. That is, the values for the X column and the Y column must form a complete, rectangular grid of bins. Input data with non-binned columns should be preprocessed with the KDE procedure (SAS/STAT), which enables you to set the number of bins for X and Y. Alternatively, the data can be preprocessed with a technique similar to the example, where a pre-defined bin width is used.

The bounding cube can be titled, rotated, and zoomed to provide a different viewpoint. By default, the outline of the bounding cube is displayed and the viewing rotation angle is 57 degrees, the tilt angle is 20 degrees and the zoom factor is 1. See the CUBE=, ROTATE=, TILT=, and ZOOM= options of the LAYOUT OVERLAY3D statement for information about how to change the viewpoint.

The X axis, Y axis, and Z axis are linear by default. You can change axis properties with the XAXISOPTS=, YAXISOPTS=, and ZAXISOPTS= options of the LAYOUT OVERLAY3D statement.

Note: When BINAXIS=TRUE, some axis options for the X- and Y-axes might not apply.

Example: BIHISTOGRAM3DPARM Statement

The following graph was generated by the "Example Program" on page 324:
**Example Program**

```sas
proc template;
   define statgraph bihistogram;
      begingraph;
         entrytitle "Distribution of Height and Weight";
         entryfootnote halign=right "SASHELP.HEART";
         layout overlay3d / cube=false zaxisopts=(griddisplay=on);
            bihistogram3dparm x=height y=weight z=count / display=all;
         endlayout;
      endgraph;
   end;
end;
run;

data heart;
   set sashelp.heart(keep=height weight);
   if height ne . and weight ne .;
      height=round(height,5);
      weight=round(weight,25);
run;

proc summary data=heart nway completetypes;
   class height weight;
   var height;
   output out=stats(keep=height weight count) N=Count;
run;

proc sgrender data=stats template=bihistogram;
run;
```

**BLOCKPLOT Statement**

Creates one or more strips of rectangular blocks containing text values. The width of each block corresponds to specified numeric intervals.

**Syntax**

```sas
BLOCKPLOT X=column | expression
BLOCK=column | expression </options>;
```

**Summary of Optional Arguments**

**Appearance options**

- `ALTFILLATTRS=style-element | style-element (fill-options) | (fill-options)`
  - specifies the appearance of alternate fills.
- `BLOCKINDEX=positive-integer-column | expression`
  - specifies indices for remapping the assignment of the color of the block fills.
- `CLASS=column | expression`
creates a separate block plot for each unique value of the specified column or expression.

**DATATRANSPARENCY=number**
specifies the degree of the transparency of the block fill and outline.

**DISPLAY=(display-options) | STANDARD | ALL**
specifies which block features to display.

**EXTENDBLOCKONMISSING=TRUE | FALSE**
specifies whether a missing value in the BLOCK column starts a new block or reverts to the previous nonmissing value.

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the block fills.

**FILLTYPE=MULTICOLOR | ALTERNATE**
specifies how the blocks are filled.

**INCLUDEMISSINGCLASS=TRUE | FALSE**
specifies whether missing values in the class column are included in the plot.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**
specifies the appearance of the block outlines.

**REPEATEDVALUES=TRUE | FALSE**
specifies whether contiguous block values that are identical create separate blocks.

### Axes options

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

### Label options

**BLOCKLABEL=column | expression**
specifies alternative text to display for the internal block text values.

**LABEL="string"**
specifies an external label for a single block plot.

**LABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the external block label(s).

**LABELPOSITION=LEFT | RIGHT | TOP | BOTTOM**
specifies the alignment of BLOCK label.

### Plot reference options

**NAME="string"**
assigns a name to this plot statement for reference in other template statements.

### Text options

**VALUEATTRS=style-element | style-element (text-options) | (text-options)**
specifies the appearance of the internal block text values or the alternative values that are specified by the BLOCKLABEL= option.

**VALUEFITPOLICY=NONE | SHRINK | SPLIT | SPLITALWAYS | TRUNCATE**
specifies how text values are adjusted to fit within the containing block.

**VALUEHALIGN=LEFT | CENTER | RIGHT | START**
specifies the horizontal alignment of the value text within the blocks.

**VALUESPLITCHAR="character-list"**
specifies one or more characters on which the values can be split, if needed.

VALUESPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the displayed values.

VALUEVALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment of the value text within the blocks.

Required Arguments

\textbf{X=column | expression}

specifies numeric X-axis positions. If the X axis is discrete and the specified column is numeric, values are treated as numeric-discrete. If the X axis is interval and the specified X column is numeric, the following guidelines apply:

- If the data is a SAS data set, you must sort the data in ascending order by the specified X column before you plot it.
- If the data is a CAS in-memory table, you do not need to sort the data. The data is automatically sorted in ascending order by the X column before it is plotted.

\textbf{BLOCK=column | expression}

specifies a value for each X position. Numeric values are converted to text strings according to an assigned format or according to BEST6.

Optional Arguments

\textbf{ALTFILLATTRS=style-element | style-element (fill-options) | (fill-options)}

specifies the appearance of alternate fills. This option in conjunction with the \texttt{FILLATTRS=} option controls fill appearance when \texttt{FILLTYPE=}ALTERNATE.

\textbf{Default}

The GraphAltBlock style element.

\textbf{Requirement}

\texttt{FILLTYPE=} ALTERNATE must set for this option to have any effect.

\textbf{Interaction}

For this option to have any effect, the fill must be enabled by the ODS style or the \texttt{DISPLAY=} option.

\textbf{Note}

If transparency is in effect with fill and the template includes a DISCRETELEGEND statement with the AUTOALIGN= option, the block plot bands might appear darker than normal. In that case, replace the AUTOALIGN= option in the DISCRETELEGEND statement with HALIGN= or VALIGN=.

\textbf{Tips}

The \texttt{FILLATTRS=} option controls the fill color.

To make all block fill areas the same color, set the \texttt{FILLATTRS=} and \texttt{ALTFILLATTRS=} options to the same value.

The \texttt{DATATRANSPARENCY=} option sets the transparency for the block fills and the outlines. You can combine this option with \texttt{DATATRANSPARENCY=} to set one transparency for the outlines but a different transparency for the alternate block fills. Example:

\texttt{datatransparency=0.2 altfillattrs=(transparency=0.6)}

\textbf{See}

“General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}. 

"Fill Color Options" on page 1520 for available fill-options.

**BLOCKINDEX=**positive-integer-column | expression

specifies indices for remapping the assignment of the color of the block fills.

**Requirements**

FILLTYPE= MULTICOLOR must be set for this option to have any effect.

The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indices for a specific block value must be the same. Otherwise, the results are unpredictable.

**Notes**

The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style element to use.

If this option is not used, then the color values are mapped in the order of GraphData1–GraphDataN.

**BLOCKLABEL=**column | expression

specifies alternative text to display for the internal block text values.

**Default**

The BLOCK= values

**Interaction**

The DISPLAY= option must include VALUES for the alternative text to appear.

**Note**

The text for each block segment must be the same. Otherwise, the results are unpredictable.

**Tips**

This option is particularly useful for showing regimes in forecasting and time series plots.

The font and color attributes for the alternative text are specified by the VALUEATTRS= option.

**CLASS=**column | expression

creates a separate block plot for each unique value of the specified column or expression. Each block plot is labeled externally by the class value.

**Interactions**

The DISPLAY= option must include LABEL for any external labels to appear.

This option overrides the LABEL= option.

**Note**

If you specify a column in a SAS data set, the visual attributes for each class value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each class value are assigned in ascending order of the class column character values or of unformatted numeric values.
Tip

The font and color attributes for the external labels are specified by the LABELATTRS= option.

**DATATRANSPERAncy=number**

specifies the degree of transparency of the block fill and outline.

| Default | 0 |
| Range   | 0–1, where 0 is opaque and 1 is entirely transparent |
| Notes   | This option does not affect the block values or labels. |

If transparency is in effect with fill and the template includes a DISCRETELEGEND statement with the AUTOALIGN= option, the block plot bands might appear darker than normal. In that case, replace the AUTOALIGN= option in the DISCRETELEGEND statement with HALIGN= or VALIGN=.

Tip

The ALTFILLATTRS= option can be used to set transparency for just the alternate block fills. The FILLATTRS= option can be used to specify transparency for the block fills. You can combine this option with ALTFILLATTRS= and with FILLATTRS= to set one transparency for the outlines but a different transparency for the block fills. Example:

```
datatransparency=0.2
altfillattrs=(transparency=0.6) fillattrs=(transparency=0.6)
```

**DISPLAY=(display-options) | STANDARD | ALL**

specifies which block features to display.

(display-options)

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

- **FILL**
  - displays the block fill color
- **LABEL**
  - displays the external block label or labels
- **OUTLINE**
  - displays the block outline
- **VALUES**
  - displays internal block values or the alternative block values that are specified by the BLOCKLABEL= option

**STANDARD**

specifies FILL and OUTLINE

**ALL**

specifies all features: FILL, LABEL, OUTLINE, and VALUES

Default | STANDARD

Tips

Use the OUTLINEATTRS=, FILLATTRS=, ALTFILLATTRS=, and BLOCKINDEX= options to control the appearance of the blocks.
Use the VALUEATTRS= and LABELATTRS= options to control the text appearance.

EXTENDBLOCKONMISSING=TRUE | FALSE
specifies whether a missing value in the BLOCK column starts a new block or reverts to the previous nonmissing value.

Default FALSE

Tip When EXTENDBLOCKONMISSING=TRUE, you can set up the input data for the BLOCK= column with nonmissing values where you expect the blocks to change and leave the remaining block values missing. For an example, see “Example 1: BlockPlot Overlaid with SeriesPlot” on page 337.

See “boolean ” on page 1479 for other Boolean values that you can use.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the block fills. This option in conjunction with the ALTFILLATTRS= option controls fill appearance when FILLTYPE= ALTERNATE.

Default The GraphBlock style element.

Requirement FILLTYPE= ALTERNATE must set for this option to have any effect.

Interaction For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Note If transparency is in effect with fill and the template includes a DISCRETELEGEND statement with the AUTOALIGN= option, the block plot bands might appear darker than normal. In that case, replace the AUTOALIGN= option in the DISCRETELEGEND statement with HALIGN= or VALIGN=.

Tips The ALTFILLATTRS= option controls the alternating fill color.

To make all block fill areas the same color, set the FILLATTRS= and ALTFILLATTRS= options to the same value.

The DATATRANSPARENCY= option sets the transparency for the block fills and the outlines. You can combine this option with TRANSPARENCY= in (fill-options) to set one transparency for the outlines and a different transparency for the block fills. For example:

datatransparency=0.2 fillattrs=(transparency=0.6)

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

FILLTYPE=MULTICOLOR | ALTERNATE
specifies how the blocks are filled.
MULTICOLOR
fills the blocks with the color specified by the COLOR attribute of the style elements GraphData1–GraphDataN and GraphMissing (for missing BLOCK= values), or the style elements indicated by the BLOCKINDEX= column.

See “BLOCKINDEX=positive-integer-column | expression” on page 327

ALTERNATE
alters the block fill between the colors specified by the FILLATTRS= and ALTFILLATTRS= options.

See “FILLATTRS=style-element | style-element (fill-options) | (fill-options)” on page 329 and “ALTFILLATTRS=style-element | style-element (fill-options) | (fill-options)” on page 326

Default MULTICOLOR

Interactions For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

When this option is set to ALTERNATE, the block plot does not support a DISCRETELEGEND entry.

INCLUDEMISSINGCLASS=TRUE | FALSE
specifies whether missing values in the class column are included in the plot. Missing class values are included by default. When the data contains missing class values, the label for those values is either blank for missing character values or a dot for missing numeric values.

The following figure shows block plots for classes Class 1, Class 2, and any missing class values.

The following figure shows an X-axis axis table that displays values for classes Class 1, Class 2, and any missing class values.

Notice that the label for the missing class values is blank. You can use the INCLUDEMISSINGCLASS=FALSE option to exclude the missing class values. If you want to keep the missing class values, then you can create a format that specifies a more meaningful label for the missing class. For example, here is a format that specifies a label for missing character and numeric class values.

```proc format;
  value $missingClass " " = "(Missing)";
  value missingClass . = "(Missing)";
run;
```

A single space enclosed in quotation marks specifies a missing character value and a dot specifies a missing numeric value. Although it might seem appropriate to use empty quotation marks ("" or ") to specify a missing character value, doing so produces unexpected results. To specify a missing character value, enclose a single space in quotation marks (" "). You can use this format for the class columns in the PROC SGRRENDER statement. In that case, if the class columns
contain missing values, then the labels specified in the format statement are used for the missing classes.

The following figure shows the previous example when format $missingClass is applied to the class variable.

![Figure showing class variables with missing values](image)

**Note:** In **SAS 9.4M2** and in earlier releases, ODS Graphics does not support Unicode values in user-defined formats. Starting with **SAS 9.4M3**, ODS Graphics supports Unicode values in user-defined formats only if they are preceded by the (**ESC**) escape sequence. Example: "(*ESC*){unicode beta}". ODS Graphics does not support an escape character that is defined in an ODS ESCAPECHAR statement in user-defined formats.

---

**Default**

**TRUE**

**Interaction**

The CLASS= option must be specified for this option to have any effect.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**LABEL=** "string"
specifies an external label for a single block plot.

**Defaults**

The label of the BLOCK= column.

The name of the BLOCK= column, if there is no column label.

**Interactions**

If the CLASS= option is specified, then this option is ignored.

The DISPLAY= option must include LABEL for any external label(s) to appear.

**Tip**

The font and color attributes for the external label are specified by the LABELATTR= option.

**LABELATTR=** style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the external block label(s).

**Default**

The GraphLabelText style element.

**Interaction**

If one or more text options are specified and they do not include all of the font properties such as color, family, size, weight, style, then the non-specified properties are derived from the GraphLabelText style element.

**See**

"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Text Options" on page 1525 for available text-options.
LABELPOSITION=LEFT | RIGHT | TOP | BOTTOM
specifies the alignment of BLOCK label.

LEFT
places the label to the left of the strip of block values.

RIGHT
places the label to the right of the strip of block values.

TOP
places the label above the strip of block values.

BOTTOM
places the label below the strip of block values.

Default LEFT

NAME="string"
assigns a name to this plot statement for reference in other template statements. This option is used mostly in the DISCRETELEGEND statement in order to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the block outlines.

Default The GraphOutlines style element.

Interactions For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

If labels are displayed in the TOP or BOTTOM position, then they are also outlined.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

REPEATEDVALUES=TRUE | FALSE
specifies whether contiguous block values that are identical create separate blocks.

FALSE
creates only one block when two or more identical block values appear consecutively.

TRUE
creates one block for each identical value when two or more identical values appear consecutively.

Default FALSE

See “boolean ” on page 1479 for other Boolean values that you can use.
VALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the appearance of the internal block text values or the alternative values that are specified by the BLOCKLABEL= option.

Default
The GraphValueText style element.

Interaction
If one or more text options are specified and they do not include all of the font properties such as color, family, size, weight, style, then the non-specified properties are derived from the GraphValueText style element.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

VALUEFITPOLICY=NONE | SHRINK | SPLIT | SPLITALWAYS | TRUNCATE
specifies how text values are adjusted to fit within the containing block.

NONE
makes no attempt to fit values that collide.

SHRINK
reduces the font size of the values until they all fit.

SPLIT
splits a value that does not fit within the containing block at a split character in order to make the value fit the available space.

No split occurs at split characters that occur where a split is not needed. If the value does not contain any of the specified split characters, then a split does not occur.

Interaction
The VALUESPLITCHAR= option specifies the split characters.

Tip
Use the VALUEHALIGN= option to control the alignment of the split lines.

SPLITALWAYS
splits each value at every occurrence of a split character.

If the value does not contain any of the specified split characters, then a split does not occur.

Interaction
The VALUESPLITCHAR= option specifies the split characters.

Tip
Use the VALUEHALIGN= option to control the alignment of the split lines.

TRUNCATE
truncates any value that does not fit.

For a numeric column, an asterisk (*) is substituted for the entire value whenever truncation occurs. For a character column, the truncated portion of the text is replaced by an ellipsis (...).

Default
TRUNCATE

Interaction
The SPLIT and SPLITALWAYS policies are ignored when the BLOCKPLOT statement is placed in a DATALATTICE or
DATAPANEL layout and the BLOCKPLOT CLASS= option is set. In that case, the TRUNCATE fit policy is used instead.

**VALUEALIGN=LEFT | CENTER | RIGHT | START**

specifies the horizontal alignment of the value text within the blocks.

- **LEFT**  
  left-aligned within the block

- **CENTER**  
  center-aligned within the block

- **RIGHT**  
  right-aligned within the block

- **START**  
  center-aligned at the starting value of the block

**Default**  
LEFT

**Restriction**  
When the BLOCKPLOT statement is placed inside an INNERMARGIN block, only VALUEALIGN=CENTER is honored.

**Requirement**  
For this option to have any effect, the DISPLAY= option must include VALUE.

**Interaction**  
When REPEATEDVALUES= TRUE and X values are numeric, only CENTER and START can be used for a discrete axis. In addition, only LEFT and START can be used for a linear or log axis. For example, if REPEATEDVALUES=TRUE and the axis is discrete, then a setting of RIGHT for this option is ignored and LEFT is used instead.

**VALUESPLITCHAR="character-list"**

specifies one or more characters on which the values can be split, if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the value. In that case, all of the specified split characters together are treated as a single split character.

When VALUESPLITPOLICY=SPLIT, if a value collision is detected, then the value is split at each occurrence of any of the specified split characters. When VALUESPLITPOLICY=SPLITALWAYS, the value is split unconditionally on each of the specified split characters. If the value does not contain any of the specified split characters, then the value is not split.

"character-list"  
one or more characters with no delimiter between each character and enclosed in quotation marks.

**Default**  
A blank space

**Requirements**  
The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no delimiters. For example, to specify the split characters a, b, and c, use the following option:

valuesplitchar="abc"
The VALUEFITPOLICY= option must specify SPLIT or SPLITALWAYS.

Interactions

The VALUESPLITCHARDROP= option specifies whether the split characters are included in the displayed value or are dropped.

The VALUEFITPOLICY= option sets the policy that is used to manage the split behavior of the value.

Notes

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

VALUESPLITCHARDROP=TRUE | FALSE

specifies whether the split characters are included in the displayed values.

TRUE

drops a split character from the value when a split occurs at that character.

Split characters at which a split does not occur are left in place. The VALUEFITPOLICY= option determines where the values are split. When VALUEFITPOLICY=SPLIT, each value is split at a split character only where a split is needed in order to make the value fit the available space. At each split point, the split character is dropped, and the characters that follow the split character, up to but not including the split character at the next split point, are wrapped to the following line.

When VALUEFITPOLICY=SPLITALWAYS, each value is split at every instance of a split character. All of the split characters are dropped. The characters that follow each split character, up to but not including the next split character, are wrapped to the next line.

The following figure shows how value Product*Group*1 is split when the VALUESPLITCHARDROP=TRUE, VALUESPLITCHAR=**, and VALUEHALIGN=CENTER options are specified with the SPLIT and SPLITALWAYS fit policies.

![Figure showing value split examples]

In this example, when VALUEFITPOLICY=SPLIT, the value is split at the first occurrence of the asterisk in order to make the value fit. No split is needed at the second asterisk. The first asterisk is dropped, and Group*1 wraps to the next line. Notice that the second asterisk is not dropped in this case. When VALUEFITPOLICY=SPLITALWAYS, the value is split at every occurrence of the asterisk. In this case, both asterisks are dropped, and the characters that follow each asterisk wrap to the next line.

FALSE

includes the split characters in the value display.

The VALUEFITPOLICY= option determines how the split characters are displayed. When VALUEFITPOLICY=SPLIT, each value is split at a split character only where a split is needed in order to make the value fit the
available space. A split might not occur at every split character in the value. At each split point, the split character remains as the last character in the current line. The characters that follow the split character, up to and including the split character at the next split point, are then wrapped to the following line. This process repeats until the entire value is displayed.

When VALUEFITPOLICY=SPLIT, each value is split at every instance of a split character in the value regardless of whether a split is actually needed. Each split character remains as the last character in the current line. The characters that follow each split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows how value Product*Group*1 is split when the VALUEFITPOLICY=SPLIT and VALUEVALIGN=CENTER options are specified with the SPLIT fit policy.

In this example, when VALUEFITPOLICY=SPLIT, the label is split at the first occurrence of the asterisk to make the value fit. No split is needed at the second asterisk. The characters that follow the first asterisk wrap to the next line. When VALUEFITPOLICY=SPLITALWAYS, the value is split at every occurrence of the asterisk. Each asterisk remains as the last character in the current line, and the characters that follow are wrapped to the next line.

Default TRUE. A split character is dropped from the value display when a split occurs at that character.

Requirement The VALUEFITPOLICY= option must specify SPLIT or SPLITALWAYS.

Interaction The VALUESPLITCHAR= option specifies the split characters.

See "boolean " on page 1479 for other Boolean values that you can use.

VALUEVALIGN=TOP | CENTER | BOTTOM specifies the vertical alignment of the value text within the blocks.

Default CENTER

Interaction For this option to have any effect, the DISPLAY= option must include VALUE.

XAXIS=X | X2 specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.
Details
A block plot contains information about X locations and text values to be associated with corresponding intervals along the X axis. There is no Y-axis information in this plot.

Examples:

Example 1: BlockPlot Overlaid with SeriesPlot
When overlaid with plots that have a Y axis (a series plot for example), a block plot expands vertically to fill the Y-axis range. In the BLOCKPLOT statement, the BLOCK= argument can be used to reference text values from a column. The resulting graph displays those values within the plot wall. This example shows how a block plot can be specified with a series plot within an OVERLAY layout. Here is the output for this example.

Example Output

![Graph showing Microsoft Share Prices and Significant OS Releases](image)

Input Data
To prepare input data for the graph, “event” information must be added to the existing data for stock prices. Here is the SAS code that generates the data for this example. Notice that the first DATA step creates a Release column. That column is later specified on the BLOCK= argument to display text values on the wall of the block plot.
data MSevents;
  input Date date9. Release $5.;
  label Release="Windows Release";
datalines;
  01jun1990 3.0
  01sep1995 95
  01jul1998 98
  01mar2000 2000
  01nov2001 XP
;
proc sort data=sashelp.stocks(keep=date stock close)
   out=MSstock;
   where stock="Microsoft";
   by date;
run;

data events;
  merge MSstock MSevents;
  by date;
run;

In the merged input data set, the Release column value is missing for each
observation until the first event defined in the MSEvents data, 3.0, occurs in June of
1990. The Release column value is missing again for the subsequent observations
until the next event defined in the MSEvents data, 95, occurs in September of 1995.
Here is a sample of the merged data.

<table>
<thead>
<tr>
<th>Obs</th>
<th>Stock</th>
<th>Date</th>
<th>Close</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>Microsoft</td>
<td>02APR90</td>
<td>$58.00</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>Microsoft</td>
<td>01MAY90</td>
<td>$73.00</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>Microsoft</td>
<td>01JUN90</td>
<td>$76.00</td>
<td>3.0</td>
</tr>
<tr>
<td>48</td>
<td>Microsoft</td>
<td>02JUL90</td>
<td>$66.50</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Microsoft</td>
<td>01AUG90</td>
<td>$61.50</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Microsoft</td>
<td>04SEP90</td>
<td>$63.00</td>
<td></td>
</tr>
</tbody>
</table>

Example Program

Here is the code for this example.

proc template;
  define statgraph blockplot1;
  begingraph;
    entrytitle "Microsoft Share Prices";
    entrytitle "and Significant OS Releases";
    layout overlay;
      blockplot x=date block=release /
        datatransparency=0.3 valuealign=top
        labelposition=top display=(fill values label)
        extendblockonmissing=true;
      seriesplot x=date y=close;
    endlayout;
  endgraph;
Details

In the GTL template code for this example, BLOCK=RELEASE is specified in the BLOCKPLOT statement so that the RELEASE values are displayed on the wall of the resulting block plot. In this template, the BLOCKPLOT statement sets EXTENDBLOCKONMISSING=TRUE so that missing values in the data revert to the previous nonmissing value in the block plot. Thus, in the block plot, values are missing until 01JUN90, when the value changes from missing to 3.0. The block plot retains that 3.0 value for subsequent observations until the next nonmissing value replaces it (in this case, the value 95 on 01SEP95). In the example output shown in “Example Output” on page 337, the fill color for each of the nonmissing values is determined by the style elements GraphData1–GraphData5 in this case. The fill color for the missing values is determined by the next available GraphDataN style element, which is GraphData6 in this case.

Example 2: Stand-Alone BlockPlot in Lattice Layout

When used as a stand-alone plot in a LATTICE layout, the block plot’s height can be controlled. Using this technique, it is possible to include two or more “event” strips in a plot. This example demonstrates this technique by creating a series plot with an event strip as shown in the following figure.
Example Output

Example Program

In this example, the lattice ROWWEIGHTS=(0.04 0.96) option apportions 4% of the vertical space to the block plot. Here is the code for this example.

```sas
proc template;
define statgraph blockplot2;
begingraph;
  entrytitle "Microsoft Share Prices";
  entrytitle "and Significant OS Releases";
  layout lattice / rowweights=(0.04 0.96);
  blockplot x=date block=release / datatransparency=0.3 valuefitpolicy=shrink labelposition=left display=(fill label outline values) extendblockonmissing=true;
  seriesplot x=date y=close;
endlayout;
endgraph;
end;
run;

proc sgrender data=events template=blockplot2;
  format date year4.;
  label date="Year";
run;
```
BOXPLOT Statement

Creates box plots that are computed from input data.

Tip: Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

BOXPLOT Y=numeric-column | expression </options>;
BOXPLOT X=column | expression
Y=numeric-column | expression <option(s)>;

Summary of Optional Arguments

Appearance options

BOXWIDTH=number
  specifies the width of a box as a ratio of the maximum possible width.

CAPSCALE=positive-number
  specifies a positive number to use as a multiplier for determining the default width of the whisker caps.

CAPSHAPE=SERIF | LINE | BRACKET | NONE
  specifies the shape at the ends of the whiskers.

CLUSTERWIDTH=number
  specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
  specifies that a connect line joins a statistic from box to box.

CONNECTATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the lines connecting multiple boxes.

CONNECTBREAK=TRUE | FALSE
  specifies whether the connect line is broken for values that have no observations.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of the filled boxes.

DATATRANSPARENCY=number
  specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, outliers, and data labels, if displayed.

DISPLAY=(display-options) | STANDARD | ALL
  specifies which box-plot features to display.

EXTREME=TRUE | FALSE
  specifies whether the whiskers can extend beyond the fences.
FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the interior fill area of the boxes.

FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

INDEX=positive-integer-column | expression
specifies indices for mapping box attributes (fill and outline) to one of the GraphData1–GranphDataN style elements.

INTERVALBOXWIDTH=AUTO | dimension
specifies the box width when an interval category (X) column is specified.

MEANATTRS=style-element | style-element (marker-options) | (marker-options)
specifies the attributes of the marker representing the mean within the box.

MEDIANATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the line representing the median within the box.

ORIENT=VERTICAL | HORIZONTAL
specifies the orientation of the Y axis and of the boxes.

OUTLIERATTRS=style-element | style-element (marker-options) | (marker-options)
specifies the attributes of the markers representing the outliers.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the box outline.

SPREAD=TRUE | FALSE
specifies whether outliers with the same value are spread out to avoid overlap.

WHISKERATTRS=style-element | style-element (line-options) | (line-options)
specifies the line properties of the whiskers and caps.

Axes options

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

OUTLIERTIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over an outlier.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns.

TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options
DATALABEL=column
specifies the labels of the outliers. Either a numeric or a character column can be used.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the outlier labels.

DATALABELSPLIT=TRUE | FALSE
specifies whether to split the data labels at the specified split characters.

DATALABELSPLITCHAR="character-list"
specifies one or more characters on which the data labels can be split if needed.

DATALABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the data labels.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the data label blocks.

LABELFAR=TRUE | FALSE
specifies whether all outliers or only far outliers are labeled.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options

DISCRETEOFFSET=number
specifies an amount to offset all boxes from the discrete X ticks.

GROUP=column | discrete-attr-var | expression
creates a box plot for each unique group value of the specified column.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how to display the boxes that represent group values for the coordinate pairs.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Plot reference options

NAME="string"
assigns a name to this plot statement for reference in other template statements.

Statistics options

DISPLAYSTATS=NONE | STANDARD | ALL | (statistics-list)
specifies the statistics to be displayed for each box.

FREQ=numeric-column | expression
specifies a numeric column that provides frequencies for each observation that is read.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies one of five definitions used to calculate percentiles.

WEIGHT=numeric-column | expression
specifies a column that contains a statistics calculation a priori weight for each observation of the input data object.

WHISKERPERCENTILE=number
specifies the whisker length, in percentile units.
Required Arguments

Specifying only Y= creates a single box plot. Specifying both X= and Y= creates a box plot for each unique value of X.

\[ Y=\text{numeric-column} \text{ | expression} \]

specifies the column for the Y values. This argument is required.

\[ X=\text{column} \text{ | expression} \]

specifies the column for the X values. This argument is required if you want to create a box plot for each unique X value.

Note For interval X values, if a user-defined format is applied to the X column, the format should map each X value to only one unique formatted value. Otherwise, unexpected results might occur.

Optional Arguments

\[ \text{BOXWIDTH=}\text{number} \]

specifies the width of a box as a ratio of the maximum possible width.

Defaults For a discrete category axis, 0.4 for a nongrouped box plot or 0.6 for a grouped box plot.

For an interval category axis, 85% of the smallest interval between any two boxes for the given plot.

Range 0–1, where 0 is the narrowest and 1 is the widest.

Interactions For a grouped box plot with a discrete category axis when \text{GROUPDISPLAY=}\text{CLUSTER}, the box width is a percentage of the cluster width. See \text{CLUSTERWIDTH}.

Prior to \text{SAS 9.4M3}, for an interval box plot, this option is ignored, and the box width is controlled by the \text{INTERVALBOXWIDTH=} option. Starting with \text{SAS 9.4M3}, for an interval box plot, this option is ignored when the \text{INTERVALBOXWIDTH=} option is specified. Otherwise, the \text{BOXWIDTH=} option is honored.

\[ \text{CAPSCALE=}\text{positive-number} \]

specifies a positive number to use as a multiplier for determining the default width of the whisker caps.

Note: This feature applies to \text{SAS 9.4M5} and to later releases.

Default 1

Interaction This option is ignored when \text{CAPSHAPE=}\text{NONE} is in effect.

Tips Use the \text{CAPSHAPE=} option to specify the cap shape.

The thickness (height) of the SERIF, LINE, and BRACKET is determined by the thickness specified in the \text{WHISKERATTRS=} option. The cap line pattern is always solid.

Examples Double the size of the caps:

\[ \text{capscale=}2 \]
Reduce the size of the caps by 50%:
\( \text{capscale}=0.5 \)

**CAPSHAPE=SERIF | LINE | BRACKET | NONE**
specifies the shape at the ends of the whiskers.

**SERIF**
specifies a short line perpendicular to the whisker.

**LINE**
specifies a line perpendicular to the whisker that extends the width of the box.

**BRACKET**
specifies a line perpendicular to the whiskers that extends the width of the box and that has short extensions at each end.

The extensions are drawn in the direction of the box.

**NONE**
specifies that no shape appears at the ends of the whiskers.

The following figure shows each of the shapes.

<table>
<thead>
<tr>
<th>SERIF</th>
<th>LINE</th>
<th>BRACKET</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="SERIF" /></td>
<td><img src="image2" alt="LINE" /></td>
<td><img src="image3" alt="BRACKET" /></td>
<td><img src="image4" alt="NONE" /></td>
</tr>
</tbody>
</table>

**Default**
The GraphBox:CapStyle style reference.

**Interactions**
The cap color and the thickness are specified by the `WHISKERATTRS=` option. The cap pattern is always solid.

The `DISPLAY=` option must include CAPS in order for cap lines to be shown.

Starting with SAS 9.4M5, when `CAPSCALE=` is in effect, the display width for the caps is the cap's default width multiplied by the value specified in `CAPSCALE=`.

**CLUSTERWIDTH=number**
specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.
Default 0.7

Range 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Requirement For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

Note When the X axis is an interval axis, the cluster width is a fraction of the smallest data interval.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
specifies that a connect line joins a statistic from box to box.

Default The GraphBox:Connect style reference.

Requirement The DISPLAY= option must contain the CONNECT display-options value in order for the connect line to be displayed.

Interaction This option applies only when the X= argument is used to generate multiple boxes.

Note Starting with SAS 9.4M3, the connect lines are drawn in axis order. In prior releases, they are drawn in data order.

CONNECTATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the lines connecting multiple boxes.

Default The GraphConnectLine style element.

Interactions If there is only one box, then this option is ignored.

If the DISPLAY= option does not include CONNECT, or if the GROUP= option is used, then this option is ignored.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
"Line Options" on page 1522 for available line-options.

**CONNECTBREAK=** **TRUE | FALSE**  
specifies whether the connect line is broken for values that have no observations.

**Default**  
FALSE

**Requirement**  
DISPLAY= must include CONNECT for this option to have any effect.

**Interaction**  
This option is ignored when the GROUP= option is in effect.

**DATALABEL=** column  
specifies the labels of the outliers. Either a numeric or a character column can be used.

**Default**  
No data labels are displayed

**Interaction**  
This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

See  
LABELFAR= option

**DATALABELATTRS=** **style-element | style-element (text-options) | (text-options)**  
specifies the color and font attributes of the outlier labels.

**Default**  
The GraphDataText style element.

**Interactions**  
This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

If one or more label options are specified and they do not include all the font properties (color, family, size, weight, style), then the non-specified properties are derived from the GraphDataText style element.

See  
"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Text Options" on page 1525 for available text-options.

**DATALABELSPLIT=** **TRUE | FALSE**  
specifies whether to split the data labels at the specified split characters. When set to TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters.

**Default**  
FALSE. The data labels are not split.

**Requirement**  
The DATALABEL= option must also be specified.

**Interaction**  
The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.
See “boolean ” on page 1479 for other Boolean values that you can use.

**DATALABELSPLITCHAR=character-list**

specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

*character-list*

one or more characters with no space between each character and enclosed in quotation marks.

**Default**

A blank space

**Requirements**

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```
datalabelsplitchar="abc"
```

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction**

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels.

**TRUE**

drops the split characters from the data label.

**FALSE**

includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:

- the data label text for this label is Product*Group*A
DATALABELSPLIT=TRUE

DATALABELSPLITCHAR=**

When DATALABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When DATALABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default

TRUE. The split characters are dropped from the data label.

Requirement

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interaction

The DATALABELSPLITCHAR= option specifies the split characters.

See

“boolean ” on page 1479 for other Boolean values that you can use.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the data label blocks.

AUTO

justifies the labels based on the DATALABELPOSITION= option.

DATALABELPOSITION= Value | Justification

<table>
<thead>
<tr>
<th>TOPLEFT, LEFT, or BOTTOMLEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

CENTER

LEFT

RIGHT

justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.

Note: The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.
In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

Default: AUTO

Requirement: The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interaction: This option has no effect if DATALABELPOSITION=AUTO.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN enhances the visual appearance of the filled boxes. The following figure shows boxes with each of the skins applied.

Default: The DATASKIN= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

Restriction: Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

Requirement: For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.
Interactions

This option overrides the BEGINGRAPH statement DATASKIN= option.

The data skin appearance is based on the FILLATTRS= color.

When a data skin is applied, all bar outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

**DATATRANSparency=number**

specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, outliers, and data labels, if displayed.

Default: 0

Range: 0–1, where 0 is opaque and 1 is entirely transparent

Tip: The FILLATTRS= option can be used to set transparency for just the interior fill area of the boxes. You can combine this option with FILLATTRS= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

datatransparency=0.2 fillattrs=(transparency=0.6)

**DISCRETEOFFSET=number**

specifies an amount to offset all boxes from the discrete X ticks.

Default: 0 (no offset, all boxes are centered on the discrete ticks)

Range: -0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is to the right when ORIENT=VERTICAL, and up when ORIENT=HORIZONTAL. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

Restriction: This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

Tip: Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

See: “About the DISCRETEOFFSET= Option” on page 367

**DISPLAY=(display-options) | STANDARD | ALL**

specifies which box-plot features to display.

*(display-options)* specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**CAPS** displays caps at the ends of the whiskers

**CONNECT** displays the line connecting multiple boxes
FILL displays the box fill color

FILLPATTERN displays the box fill pattern

**Note** This feature applies to SAS 9.4M5 and to later releases.

**Tip** By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the FILLPATTERNATTRS= option to specify a fill pattern for the boxes.

MEAN displays the mean symbol within the box

MEDIAN displays the median line within the box

NOTCHES displays notched boxes

OUTLIERS displays markers for the outliers

STANDARD specifies CAPS, FILL, MEAN, MEDIAN, and OUTLIERS

ALL specifies all features: CAPS, CONNECT, FILL, FILLPATTERN, MEAN, MEDIAN, NOTCHES, and OUTLIERS

The endpoints of the notches are at the following computed locations:

\[
\text{median} \pm \frac{1.58 \times \text{IQR}}{\sqrt{n}}
\]

In the equation, IQR (IQR=Q3-Q1) is the interquartile range and \( n \) is the sample size. The medians (central lines) of the two boxes are significantly different at approximately the 0.05 level if the corresponding notches do not overlap.
### Default
The GraphBox:DisplayOptts style reference. If this style element does not exist, then the default is STANDARD.

### Interaction
If EXTREME= TRUE, then the OUTLIERS feature is ignored

### Notes
Starting with **SAS 9.4M3**, connect lines are drawn in axis order. They are drawn in data order in prior releases.

Starting with **SAS 9.4M3**, when DISPLAY= includes MEAN, the BOXPLOT statement contributes its mean markers to a discrete legend when TYPE=MARKER is in effect in the DISCRETELEGEND statement.

### Tips
To control the appearance of these features, use the CONNECTATTRS=, FILLATTRS=, MEANATTRS=, MEDIANATTRS=, OUTLIERATTRS=, and WHISKERATTRS= options. The WHISKERATTRS= option controls affects both CAPS and WHISKERS. Starting with **SAS 9.4M5**, use the FILLPATTERNATTRS= on page 355 option to control the appearance of the box fill pattern.

Regardless of which display options are being displayed, this option does not affect the axis range.

**DISPLAYSTATS=NONE | STANDARD | ALL | (statistics-list)**
specifies the statistics to be displayed for each box.

**NONE**
does not display any statistics.

**STANDARD**
displays N, MEAN, and STD.

**ALL**
displays all available statistics.

**(statistics-list)** specifies a space-separated list of one or more statistics, enclosed in parentheses.

The list can include one or more of the following:

**DATAMAX**
maximum data value that includes not only the maximum whisker values but also the maximum outlier values.

This option is valid in **SAS 9.4M1** and later releases. The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.

**DATAMIN**
minimum data value that includes not only the minimum whisker values but also the minimum outlier values.

This option is valid in **SAS 9.4M1** and later releases. The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.

**IQR**
interquartile range (Q3–Q1).
MAX
maximum data value below the box upper fence.

MEAN
mean data value for the box.

MEDIAN
median data value for the box.

MIN
minimum data value above the box lower fence.

N
number of observations for the box.

Q1
lower quartile (25th percentile) for the box.

Q3
upper quartile (75th percentile) for the box.

RANGE
range of the data (MAX–MIN).

STD
standard deviation of the data for the box.

SUMWGT
sum of the weights for the box.

This option is valid in SAS 9.4M1 and later releases.

Default: NONE

Restrictions: Prior to SAS 9.4M5, this option is ignored if ORIENT=HORIZONTAL. Starting with SAS 9.4M5, statistics can be displayed for both vertical and horizontal box plots.

This option is ignored if the GROUP= option is also specified.

Note: The notches in the box plot can extend beyond DATAMIN and DATAMAX in some cases.

EXTREME=TRUE | FALSE
specifies whether the whiskers can extend beyond the fences. Fences are locations above and below the box. The upper and lower fences are located at a distance 1.5 times the Interquartile Range (IQR) (IQR = Q3 - Q1). The upper and lower far fences are located at a distance 3 times the IQR.

FALSE
specifies that whiskers be drawn from the upper edge of the box to the largest value within the upper fence, and from the lower edge of the box to the smallest value within the lower fence.

This representation is sometime called a schematic box plot or a Tukey box plot.

TRUE
specifies that whiskers be drawn to the largest and smallest data values, whether these values are inside or outside the fences.

The outliers and far outliers are not displayed and are not labeled. This representation is sometime called a skeletal box and whisker plot.
Default: FALSE

Interaction: This option overrides the `DATALABEL=`, `DATALABELATTRS=`, `LABELFAR=`, `OUTLIERATTRS=`, and `SPREAD=` options.

See: “Statement Summary” on page 366

“boolean” on page 1479 for other Boolean values that you can use.

**FILLATTRS=** `style-element | style-element (fill-options) | (fill-options)`

specifies the appearance of the interior fill area of the boxes.

Defaults: For non-grouped data, the GraphDefault:Color style reference.

For grouped data, the Color attribute of the GraphData1–GraphDataN style elements.

Interaction: For this option to have any effect, the fill must be enabled by the ODS style or the `DISPLAY=` option.

Tip: The `DATATRANSPARENCY` option sets the transparency for the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers. You can combine this option with `DATATRANSPARENCY=` to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

See: “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Fill Color Options” on page 1520 for available `fill-options`.

**FILLPATTERNATTRS=** `style-element | (fill-pattern-options)`

specifies the appearance of the pattern-filled areas.

Note: This feature applies to SAS 9.4M5 and to later releases.

Defaults: For non-grouped data, the ContrastColor and FillPattern attributes of GraphDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

Interaction: For this option to take effect, the `DISPLAY=` option must include `FILLPATTERN` among the display options.

Tip: In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify `ATTRPRORITY=NONE` in an ODS GRAPHICS statement or in the `BEGINGRAPH` statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

See: “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`. 
“Fill Pattern Options” on page 1521 for available fill-pattern-options.

**FREQ=numeric-column | expression**
specifies a numeric column that provides frequencies for each observation that is read.

- **Default**: All observations have a frequency count of 1.
- **Restriction**: If the value of the numeric-column is missing or is less than 1, then the observation is not used in the analysis. If the value is not an integer, then only the integer portion is used.
- **Note**: If $n$ is the value of the numeric column for a given observation, then that observation is used $n$ times for the purposes of any statistical computation.

**GROUP=column | discrete-attr-var | expression**
creates a box plot for each unique group value of the specified column.

- **discrete-attr-var**: specifies a discrete attribute map variable.
  - The discrete attribute map variable is created by a DISCRETEATTRVAR statement.
  - **Restriction**: A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

A distinct box is created for each group value by varying the visual attributes of the box display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The Color attribute controls the fill color, and the ContrastColor attribute controls the outline color. The box outline line style is solid.</td>
</tr>
<tr>
<td>Fill pattern 2</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern and box outline. The box outline line style is solid. If the active ODS style does not specify FillPattern, fill patterns are not displayed by default.</td>
</tr>
<tr>
<td>Median, caps, and notches</td>
<td>ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style.</td>
</tr>
<tr>
<td>Display Feature</td>
<td>Style Attributes That Control Default Group Appearance</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Mean and outliers</td>
<td>MarkerSymbol and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
</tbody>
</table>

1 The default display features are determined by the DisplayOpts attribute of the GraphBox style element.

2 Fill patterns are valid for box plots starting with SAS 9.4M5.

Interactions

Connect lines are not drawn for grouped data.

The box plot display depends on the setting for the GROUPDISPLAY= option.

By default, the group values are mapped in the order of the data. The GROUPORDER= option can be used to control the sorting order of the group values. The INDEX= option can be used to alter the default sequence of colors and markers.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Tip

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the box outlines, but the PATTERN= setting in the OUTLIERATTRS= option could be used to assign the same line pattern to all box outlines and connect lines.

See

"Example 2: Box Plot of City MPG and Vehicle Type Grouped by Origin" on page 376

"DISCRETEATTRVAR Statement" on page 1435

**GROUPDISPLAY=OVERLAY | CLUSTER**

specifies how to display the boxes that represent group values for the coordinate pairs. The following example shows a box plot with GROUPDISPLAY=CLUSTER:
OVERLAY
draws boxes for a given group value at the exact coordinate.
Depending on the data, boxes at a given coordinate might overlap.

CLUSTER
draws boxes for a given group value adjacent to each other.

Default  OVERLAY

Interaction This option is ignored unless GROUP= is specified.

Tip Use the CLUSTERWIDTH= option to control the width of the clusters when CLUSTER is in effect.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in the group-column data order.

REVERSEDATA
orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip This option is useful when you want to reverse the category axis.

ASCENDING
orders the groups within a category in ascending order.

DESCENDING
orders the groups within a category in descending order.

Defaults DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.
Interactions

This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

**INCLUDEMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the group variable are included in the plot.

Default TRUE

Interaction For this option to take effect, the GROUP= option must also be specified.

Tip The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See “boolean ” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**

specifies indices for mapping box attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

Requirements The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.
### Interaction

For this option to take effect, the GROUP= option must also be specified.

### Notes

The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

### Tip

You can use indexing to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 198.

### INTERVALBOXWIDTH=AUTO | dimension

specifies the box width when an interval category (X) column is specified.

**AUTO**

the system determines the box width.

Prior to SAS 9.4M3, AUTO uses 85% of the smallest interval between any two boxes for the given plot. Starting with SAS 9.4M3, if the BOXWIDTH= option is specified, AUTO uses the BOXWIDTH= option value. Otherwise, it uses 85% of the smallest interval between any two boxes for the given plot.

uses one of the following box-width values listed in the order of precedence:

**dimension**

sets the box width to the specified value.

**Default** AUTO

**Requirement** The category axis (X by default) must be an interval axis for the INTERVALBOXWIDTH= option to have any effect.

**Interaction** Prior to SAS 9.4M3, this option controls the box width for an interval box plot. Starting with SAS 9.4M3, this option overrides the BOXWIDTH= option for an interval box plot.

**Tip** The box width that you specify with this option is honored even if the boxes overlap. If the boxes overlap, reduce the interval box width or use the BOXWIDTH= option instead.

**See** "dimension" on page 1480

### LABELFAR=TRUE | FALSE

specifies whether all outliers or only far outliers are labeled.

**FALSE**

applies the labels specified by the DATALABEL= option to both the outliers and the far outliers.

**TRUE**

applies the labels specified by the DATALABEL= option to the far outliers.

**Default** FALSE

**Interaction** This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.
See “DATALABEL=column ” on page 385

“Statement Summary” on page 366 for information about outliers.

“boolean ” on page 1479 for other Boolean values that you can use.

<table>
<thead>
<tr>
<th><strong>LEGENDLABEL=“string”</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies a label to be used in a discrete legend for this plot.</td>
</tr>
<tr>
<td><strong>Default</strong></td>
</tr>
<tr>
<td><strong>Restriction</strong></td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
</tr>
</tbody>
</table>

| **MEANATTRS=**<br>**style-element | style-element (marker-options) | (marker-options)** |
|---|
| specifies the attributes of the marker representing the mean within the box. |
| **Defaults** | For non-grouped data, GraphBoxMean style element. |
| For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1–GraphDataN style elements. |
| **Interaction** | This option is ignored if the DISPLAY= option does not display the mean. |
| **See** | “General Syntax for Attribute Options” on page 1519 for the syntax on using a **style-element**. |
| “Marker Options” on page 1523 for available **marker-options**. |

| **MEDIANATTRS=**<br>**style-element | style-element (line-options) | (line-options)** |
|---|
| specifies the appearance of the line representing the median within the box. |
| **Defaults** | For non-grouped data, the GraphBoxMedian style element. |
| For grouped data, the LineStyle and LineThickness attributes of the GraphBoxMedian style element, and the ContrastColor attribute of the GraphData1–GraphDataN style elements. |
| **Interaction** | This option is ignored if the DISPLAY= option does not display the median. |
| **See** | “General Syntax for Attribute Options” on page 1519 for the syntax on using a **style-element**. |
| “Line Options” on page 1522 for available **line-options**. |

<table>
<thead>
<tr>
<th><strong>NAME=“string”</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.</td>
</tr>
<tr>
<td><strong>Restrictions</strong></td>
</tr>
</tbody>
</table>
This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**

The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**ORIENT=VERTICAL | HORIZONTAL**

specifies the orientation of the Y axis and of the boxes.

Default  VERTICAL

**OUTLIERATTRS=style-element | style-element (marker-options) | (marker-options)**

specifies the attributes of the markers representing the outliers.

Defaults  
For non-grouped data, GraphOutlier style element.

For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1–GraphDataN style elements.

**Interaction**

This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Marker Options” on page 1523 for available marker-options.

**OUTLIERTIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over an outlier. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the box plot can be specified along with roles that do contribute.

**(role-list)**

an ordered, space-separated list of unique BOXPLOT roles. BOXPLOT roles for OUTLIERTIP include X, Y, STAT, and DATALABEL.

**Note**

In the data tip, the STAT role displays the text “outlier” or “far outlier” as applicable.

**Example**

The following example displays data tips only for the column that is assigned to the X role:

OUTLIERTIP= (X)

Default  
The columns assigned to these roles are automatically included in the data tip information: X and Y.

**Requirement**

To generate data tips, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and write the graphs to the ODS HTML destination.

**Interaction**

The labels and formats for the OUTLIERTIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

**See**

the TIP= option for specifying the information to display when the cursor is positioned over a box or whisker in the box plot.
OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the box outline.

Defaults
- For non-grouped data, the ContrastColor, LineThickness, and
  LineStyle attributes of the GraphOutlines style element.
- For grouped data and filled boxes, the LineStyle and LineThickness
  attributes of the GraphOutlines style element, and the ContrastColor
  attribute of the GraphData1–GraphDataN style elements.
- For grouped data and unfilled boxes, the LineThickness attribute of the
  GraphOutlines style element, and the ContrastColor and LineStyle
  attributes of the GraphData1–GraphDataN style elements.

See
- “General Syntax for Attribute Options” on page 1519 for the syntax on
  using a style-element.
- “Line Options” on page 1522 for available line-options.

PERCENTILE=1 | 2 | 3 | 4 | 5
specifies one of five definitions used to calculate percentiles.

Default 5 (empirical distribution function with averaging)

Note The percentile definition and default are the same as those that are
used by the PCTLDEF= option of the UNIVARIATE procedure and the
QNTLDEF= option of the SUMMARY procedure.

See “Calculating Percentiles” on page 368

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for
determining default axis features. This option is needed only when two or more
plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDED or
LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the
primary plot on a per-axis basis. When no plot is designated as the
primary plot, the first plot that can be a primary plot is considered the
primary plot. If multiple plots specify PRIMARY=TRUE for the same
axis, then the last such plot encountered is considered the primary
plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

SPREAD=TRUE | FALSE
specifies whether outliers with the same value are spread out to avoid overlap.
For vertical box plots this means offsetting the outliers horizontally. If this option
is false, then outliers with the same value are plotted in the same position, which
means only one is visible.

Default FALSE
Interaction  This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over a box or whisker in the box plot. If this option is used, then it replaces all of the information that is displayed by default.

**role-list**

an ordered, space-separated list of unique BOXPLOT roles.

BOXPLOT roles for TIP include DATAMAX, DATAMIN, MAX, MIN, MEAN, MEDIAN, N, Q1, Q2, STD, SUMWGT, and X.

**Note**  The roles DATAMAX, DATAMIN, and SUMWGT apply to SAS 9.4M1 and later releases.

**Tip**  Statistics such as N, MIN, and MAX are special roles. They are not column-based like the X role.

**Example**  The following example displays data tips only for the columns that are assigned to the roles X (CATEGORY) and the statistic MEAN:

```
TIP=(X MEAN)
```

**NONE**
suppresses data tips from the plot.

**Default**  The columns assigned to these roles are automatically included in the data tip information: X, N, STD, MIN, MAX, Q1, Q3, MEAN, and MEDIAN.

**Requirement**  To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interaction**  This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tip**  The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

**See**  the OUTLIERTIP= option for specifying the information to display when the cursor is positioned over an outlier.

**TIPFORMAT=(role-format-list)**
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

**role-format-list**

a space-separated list of role-name = format pairs.

**Example**  
```
TIP=(X Y)
TIPFORMAT=(X=4. Y=4.)
```
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

A column must be assigned to each of the specified roles.

_TIPLABEL=(role-label-list)_
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example
_TIP=(X)_
_TIPLABEL=(X="Type")_

The column label or column name of the column assigned to the role.

Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

A column must be assigned to each of the specified roles.

_WEIGHT=numeric-column | expression_
specifies a column that contains a statistics calculation a priori weight for each observation of the input data object.

The value must be nonnegative.

The values of the weight variable must be greater than zero. If the weight value for an observation is zero, negative, or missing, that observation is deleted from the analysis.

_WHISKERATTRS=style-element | style-element (line-options) | (line-options)_
specifies the line properties of the whiskers and caps.

For non-grouped data, the GraphBoxWhisker style element.

For grouped data, the LineStyle and LineThickness attributes of the GraphBoxWhisker style element, and the ContrastColor attribute of the GraphData1–GraphDataN style elements.

The caps are always drawn with a solid line.

“General Syntax for Attribute Options” on page 1519 for the syntax on using a _style-element_.

“Line Options” on page 1522 for available _line-options_.

_WHISKERPERCENTILE=number_
specifies the whisker length, in percentile units. When this option is specified, _number_ is used as the low percentile, and 100–_number_ is used as the high percentile.

Here are some examples of values and their effect:
specifies the high and low extremes

10 specifies the 10th percentile low and the 90th percentile high

25 specifies the 25th percentile low and the 75th percentile high

Default The whiskers are drawn from the box to the most extreme point that is less than or equal to 1.5 times the IQR

Range 0–25

Notes When this option is specified, fences and far outliers are not drawn.

When this option is set to 25, no whiskers are drawn because the box extends from the 25th to the 75th percentile.

**XAXIS=**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Interactions This option is ignored if the X= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=**

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default Y

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details

Statement Summary

The BOXPLOT statement displays a single box if given just a Y argument. It displays multiple boxes if given both Y and X arguments and X has more than one unique value.

By default for numeric or character columns, the category (X) axis is TYPE=DISCRETE. You can override the default and set the TYPE= to LINEAR or TIME in the parent layout, provided that the X column is numeric. The axis for the analysis (Y) column is always LINEAR. When the X axis is LINEAR, you must use the INTERVALBOXWIDTH= option to specify the box width.

When ORIENT= VERTICAL, the X (or X2) axis is used for the X column and the Y (or Y2) axis is used for the Y column. When ORIENT=HORIZONTAL, the X (or X2) axis is used for the Y column and the Y (or Y2) axis is used for the X column.

Two basic box plot representations can be drawn with the BOXPLOT statement: a schematic (Tukey) box plot and a skeletal box plot. See the EXTREME= option for details.
The following figure illustrates the box plot elements:

As shown in the figure, the bottom and top edges of the box are located at the 25th and 75th percentiles of the sample. Within the box, you can display the median (50th percentile) as a line and the mean as a marker (see `DISPLAY=` option).

You can also display markers and data labels for outliers. Outliers are observations that are more extreme than the upper and lower fences ($\pm 1.5 \times IQR$). Outliers that are beyond upper and lower far fences ($\pm 3 \times IQR$) are called FAR OUTLIERS and can also be identified and labeled. From a graphical perspective, the locations of fences along the axis are known, but there is no line or marker that displays a fence. (See `DISPLAY=`, `LABELFAR=`, and `DATALABEL=` options).

Finally, you can control the range represented by the whiskers. By default, the whiskers are drawn from the upper edge of the box to the MAX value, and from the lower edge of the box to the MIN value. (See the `EXTREME=` option.)

About the `DISCRETEOFFSET=` Option

This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BOXPLOT statements are used with different analysis variables, then the boxes for matching X values are centered on the ticks. Depending on the data, the boxes might be superimposed. The following code fragment shows the default box positioning:

```
layout overlay / cycleattrs=true
   yaxisopts=(label="Miles Per Gallon");
   boxplot x=type y=mpg_city / name="City"
      legendlabel="City";
   boxplot x=type y=mpg_highway / name="Highway"
      legendlabel="Highway";
   discretelegend "City" "Highway";
endlayout;
```
To place the different response values side by side, you can assign a different offset to each BOXPLOT statement. The BOXWIDTH= option can be used in conjunction with the DISCRETEOFFSET= option to create narrower boxes when desired.

```
layout overlay / cycleattrs=true
  yaxisopts=(label="Miles Per Gallon");
  boxplot x=type y=mpg_city / name="City"
    discreteoffset=0.2 legendlabel="City";
  boxplot x=type y=mpg_highway / name="Highway"
    discreteoffset=-0.2 legendlabel="Highway";
  discretelegend "City" "Highway";
endlayout;
```

Calculating Percentiles

You can specify one of five definitions for computing the percentiles with the PERCENTILE= option. Let \( n \) be the number of nonmissing values for a variable, and let \( X_1, X_2, \ldots, X_n \) represent the ordered values of the variable. \( X_1 \) is the smallest
value, \( X_2 \) is the next smallest, and \( X_n \) is the largest value. Let the \( r \)th percentile be \( y \), set:

\[
P = \frac{r}{100}
\]

and let:

\[
np = j + g
\]

when \( \text{PERCENTILE}=1, 2, 3, \) or \( 5 \), or let:

\[
(n + 1)p = j + g
\]

when \( \text{PERCENTILE}=4 \), where \( j \) is the integer part of \( np \), and \( g \) is the fractional part of \( np \). Then the \( \text{PERCENTILE}= \) option defines the \( r \)th percentile, \( y \), as described in the following table:

<table>
<thead>
<tr>
<th>Percentile</th>
<th>Description</th>
<th>Equation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Weighted average at ( X_{np} )</td>
<td>( y = (1 - g)x_j + gx_j + 1 )</td>
<td>( x_0 ) is taken to be ( x_1 )</td>
</tr>
<tr>
<td>2</td>
<td>Observation numbered closest to ( np )</td>
<td>( y = x_j )</td>
<td>Used when ( g &lt; \frac{1}{2} ) or</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>when ( g = \frac{1}{2} ) and ( j ) is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>even</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( y = x_j + 1 )</td>
<td>Used when ( g = \frac{1}{2} ) and ( j )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>is odd or when ( g &gt; \frac{1}{2} )</td>
</tr>
<tr>
<td>3</td>
<td>Empirical distribution function</td>
<td>( y = x_j )</td>
<td>Used when ( g = 0 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( y = x_j + 1 ) Used when ( g &gt; 0 )</td>
</tr>
<tr>
<td>4</td>
<td>Weighted average aimed at ( X_{(n+1)p} )</td>
<td>( y = (1 - g)x_j + gx_j + 1 )</td>
<td>( x_{n+1} ) is taken to be ( x_n )</td>
</tr>
<tr>
<td>5</td>
<td>Empirical distribution function with averaging</td>
<td>( y = \frac{1}{2}(x_j + x_j + 1) )</td>
<td>Used when ( g = 0 )</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( y = x_j + 1 ) Used when ( g &gt; 0 )</td>
</tr>
</tbody>
</table>

Changing Box Plot Display

SAS defines graphical style elements that control the display of box plots generated with the BOXLOT or BOXPLOTPARM statement. Using these style elements as a starting point, you can change the style attribute values to achieve a very different appearance for your box plots. Using the DEFAULT style for an example, here is a portion of the style template for elements that are related to box plots:

```sas
proc template;
  define style Default;
```

... style GraphBox / capstyle = "serif" connect = "mean" displayopts = "fill caps median mean outliers";
style GraphBoxMean / ...;
style GraphBoxMedian / ...;
style GraphBoxOutlier / ...;
style GraphBoxWhisker / ...;
...
end;
run;

Table 6.1  Style Elements and Their Purpose

<table>
<thead>
<tr>
<th>Style Element</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>GraphBox</td>
<td>general box plot properties (see the next table)</td>
</tr>
<tr>
<td>GraphBoxMean</td>
<td>marker properties of mean marker (MARKERSYMBOL= , MARKERSIZE=, CONTRASTCOLOR=)</td>
</tr>
<tr>
<td>GraphBoxMedian</td>
<td>line properties of the median line (LINESTYLE=, LINETHICKNESS=, CONTRASTCOLOR=)</td>
</tr>
<tr>
<td>GraphBoxOutlier</td>
<td>marker properties of outliers (MARKERSYMBOL=, MARKERSIZE=, CONTRASTCOLOR=)</td>
</tr>
<tr>
<td>GraphBoxWhisker</td>
<td>line properties of whiskers and caps (LINESTYLE=, LINETHICKNESS=, CONTRASTCOLOR=)</td>
</tr>
</tbody>
</table>

Table 6.2  Attributes and Values for the GraphBox Style Element

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECT=</td>
<td>&quot;MEAN&quot;</td>
<td>&quot;MEDIAN&quot;</td>
</tr>
<tr>
<td>CAPSTYLE=</td>
<td>&quot;SERIF&quot;</td>
<td>&quot;LINE&quot;</td>
</tr>
<tr>
<td>DISPLAYOPTS</td>
<td>&quot;&lt;CAPS&gt;&quot;</td>
<td>show caps at end of whiskers</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt;FILL&gt;&quot;</td>
<td>show filled boxes</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt;MEAN&gt;&quot;</td>
<td>show a marker for the mean</td>
</tr>
<tr>
<td></td>
<td>&quot;&lt;MEDIAN&gt;&quot;</td>
<td>show a line for the median</td>
</tr>
</tbody>
</table>
The DISPLAYOPTS attribute of GraphBox lists the general features to be displayed. The following diagram shows the standard display for box plots, as defined by the DEFAULT style. The keywords that are related to the appearance features are annotated:

The two display options that are not the default are CONNECT (show connect lines) and NOTCHES.
The STATISTICAL style is derived from the DEFAULT style and inherits the GraphBox element from the parent DEFAULT style. The following code generates a box plot for the STATISTICAL style:

```sas
/* Specify a path for the ODS output */
filename odsout "output-path";

proc template;
    define statgraph boxplotdef;
    begingraph;
        entrytitle "Statistical Style";
        layout overlay / xaxisopts=(label="Age" type=linear);
            boxplot x=ageatstart y=cholesterol / intervalboxwidth=40;
        endlayout;
    endgraph;
end;
end;
ods graphics / outputfmt=static;
ods _all_ close;
ods html path=odsout file="boxplot.html" style=statistical;

proc sgrender data=sashelp.heart template=boxplotdef;
    where ageatstart between 50 and 55;
run;
ods html close;
ods html; /* Not required in SAS Studio */
```

Here is the output.
For this example, the following attributes on the default box plot are changed:

- By default, serif caps are displayed at the end of the fences. Those caps are removed from the fence lines.
- By default, the boxes are filled. Empty, notched boxes are displayed.
- By default, the mean values are represented by hollow diamonds. Filled diamonds with a slightly reduced size are displayed.
- By default, the marker symbols for the outliers are hollow black circles. The size and shape of the marker symbols are changed, and the marker size is reduced.

To make these changes, you can derive a new style from the STATISTICAL style and set the attributes that you want to change. Any attribute settings that you do not change are inherited from the parent STATISTICAL style. The following style template effects the desired changes:

```
proc template;
  define style Boxplot;
  parent = styles.statistical;
  style GraphBox from GraphBox /
    capstyle = "line"
    displayopts = "caps median mean outliers notches";
  style GraphBoxMean from GraphBoxMean /
    markersymbol="diamondfilled"
    contrastcolor=GraphColors("gcdata1")
    markersize = 5px;
  style GraphOutlier from GraphOutlier /
    markersize = 5px
    markersymbol = "x"
    contrastcolor = GraphColors("gcdata2");
end;
run;
```

Note the following:
- The DEFINE STYLE statement assigns the name BOXPLOT to our new style, and sets the STATISTICAL style as the parent style.
On the GraphBox style element, the CAPSTYLE= attribute is set to LINE, which removes the serif caps from the end of the fences. The DISPLAYOPTS= attribute drops the FILL value from the display list and adds the NOTCHES value; these changes determine that the graph displays empty, notched boxes.

On the GraphBoxMean style element, the marker symbol is changed to a filled diamond and the marker size is reduced to 5 pixels (the default is 9 pixels). The CONTRASTCOLOR= attribute is set to GCDATA1 (the default is GCDATA).

On the GraphBoxOutlier style element, the marker symbol is changed to an X and the marker size is reduced to 5 pixels (the default is 7 pixels). The CONTRASTCOLOR= attribute is set to GCOUTLIER.

The following code generates a box plot for the BOXPLOT style:

```sas
/* Specify a path for the ODS output */
filename odsout "output-path";

proc template;
    define statgraph boxplotdef;
    begingraph;
        entrytitle "Boxplot Style";
        layout overlay / xaxisopts=(label="Age" type=linear);
            boxplot x=ageatstart y=cholesterol / intervalboxwidth=40;
        endlayout;
    endgraph;
end;

ods graphics / outputfmt=static;
ods _all_ close;
ods html path=odsout file="notchedboxplot.html" style=Boxplot;

proc sgrender data=sashelp.heart template=boxplotdef;
    where ageatstart between 50 and 55;
run;

ods html close;
ods html; /* Not required in SAS Studio */
```

Here is the output.
When making such style changes, remember that you are affecting all box plot displays for all procedures that produce box plots when this style is in effect. It is possible to change the box plot appearance for specific procedures, but to do this, a specific graph template must be modified, not a style template.

For a comprehensive description of the style elements affecting ODS graphics, see “Graph Style Elements Used by ODS Graphics” in SAS Graph Template Language: User’s Guide.

Examples:

Example 1: Box Plot of City MPG and Vehicle Type

The following graph was generated by the “Example Program” on page 376:
Example Program

```
proc template;
  define statgraph boxplot;
  begingraph;
    entrytitle "City Mileage for Vehicle Types";
    layout overlay /
      xaxisopts=(offsetmin=0.1 offsetmax=0.1);
    boxplot y=mpg_city x=type /
      datalabel=make spread=true;
  endlayout;
  endgraph;
run;

proc sgrender data=sashelp.cars template=boxplot;
  label type="Vehicle Type";
run;
```

Example 2: Box Plot of City MPG and Vehicle Type Grouped by Origin

The following graph was generated by the "Example Program" on page 377:
Example Program

```sas
proc template;
  define statgraph boxplot;
  begingraph;
    entrytitle "City Mileage and Vehicle Type By Origin";
    layout overlay / xaxisopts=(offsetmin=0.1 offsetmax=0.1);
    boxplot y=mpg_city x=type / name="box"
          group=origin groupdisplay=cluster
          spread=true datalabel=make;
    discretelegend "box";
  endlayout;
  endgraph;
end;
run;
proc sgrender data=sashelp.cars template=boxplot;
  label type="Vehicle Type"
run;
```

**BOXPLOTPARM Statement**

Creates side-by-side box plots specified by parameters.

**Requirements:**
- The input data must be precomputed. See "Input Data Requirements for the BOXPLOTPARM Statement" on page 407.
- Nonmissing Y values for statistical observations of Q1 and Q3 are required in order for a box to be drawn.
- The statistical values, if present, must conform to the following rules in order for a box to be displayed:
Tip: Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

BOXPLOTPARM Y=numeric-column | expression
STAT=string-column <option(s)>;

BOXPLOTPARM X=column | expression
Y=numeric-column | expression
STAT=string-column <options>;

Summary of Optional Arguments

Appearance options

BOXWIDTH=number
  specifies the width of a box as a ratio of the maximum possible width.

CAPSCALE=positive-number
  specifies a positive number to use as a multiplier for determining the
default width of the whisker caps.

CAPSHAPE=SERIF | LINE | BRACKET | NONE
  specifies the shape at the ends of the whiskers.

CLUSTERWIDTH=number
  specifies the width of the group clusters as a fraction of the midpoint
  spacing or bin width.

CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX
  specifies that a connect line joins a statistic from box to box.

CONNECTATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the lines connecting multiple boxes.

CONNECTBREAK=TRUE | FALSE
  specifies whether the connect line is broken for values that have no
  observations.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of the filled boxes.

DATATRANSPARENCY=number
  specifies the degree of the transparency of the box outlines, box fill,
  whiskers, mean, median, caps, connect lines, outliers, and data labels, if
displayed.

DISPLAY=(display-options) | STANDARD | ALL
  specifies which box-plot features to display.

EXTREME=TRUE | FALSE
  specifies whether the whiskers can extend beyond the fences.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the interior fill area of the boxes.

**FILLPATTERNATTRS=** *style-element | (fill-pattern-options)*

specifies the appearance of the pattern-filled areas.

**INDEX=** *positive-integer-column | expression*

specifies indices for mapping box attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**INTERVALBOXWIDTH=** *AUTO | dimension*

specifies the box width when an interval category (X) column is specified.

**MEANATTRS=** *style-element | style-element (marker-options) | (marker-options)*

specifies the attributes of the marker representing the mean within the box.

**MEDIANATTRS=** *style-element | style-element (line-options) | (line-options)*

specifies the appearance of the line representing the median within the box.

**ORIENT=** *VERTICAL | HORIZONTAL*

specifies the orientation of the Y axis and of the boxes.

**OUTLIERATTRS=** *style-element | style-element (marker-options) | (marker-options)*

specifies the attributes of the markers representing the outliers.

**OUTLINEATTRS=** *style-element | style-element (line-options) | (line-options)*

specifies the appearance of the box outline.

**SPREAD=** *TRUE | FALSE*

specifies whether outliers with the same value are spread out to avoid overlap.

**WHISKERATTRS=** *style-element | style-element (line-options) | (line-options)*

specifies the line properties of the whiskers and caps.

### Axes options

**PRIMARY=** *TRUE | FALSE*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

**XAXIS=** *X | X2*

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS=** *Y | Y2*

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

### Data tip options

**OUTLIERTIP=** *(role-list) | NONE*

specifies the information to display when the cursor is positioned over an outlier.

**ROLENAME=** *(role-name-list)*

specifies user-defined roles that can be used to display information in the data tips.

**TIP=** *(role-list) | NONE*

specifies the information to display when the cursor is positioned over a box or whisker in the box plot.

**TIFFORMAT=** *(role-format-list)*

specifies display formats for tip columns.

**TIPLABEL=** *(role-label-list)*

specifies display labels for tip columns.
Label options

**DATALABEL=column**
specifies the labels of the values that are identified as outlier or far outlier by the STAT= column.

**DATALABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the outlier labels.

**DATALABELSPLIT=TRUE | FALSE**
specifies whether to split the data labels at the specified split characters.

**DATALABELSPLITCHAR="character-list"**
specifies one or more characters on which the data labels can be split if needed.

**DATALABELSPLITCHARDROP=TRUE | FALSE**
specifies whether the split characters are included in the data labels.

**DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**
specifies the justification of the strings that are inside the data label blocks.

**LABELFAR=TRUE | FALSE**
specifies whether all outliers or only far outliers are labeled.

**LEGENDLABEL="string"**
specifies a label to be used in a discrete legend for this plot.

Midpoint options

**DISCRETEOFFSET=number**
specifies an amount to offset all boxes from the discrete X ticks.

**GROUP=column | discrete-attr-var | expression**
creates a box plot for each unique group value of the specified column.

**GROUPDISPLAY=OVERLAY | CLUSTER**
specifies how to display the boxes that represent group values for the coordinate pairs.

**GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING**
specifies the ordering of the groups within a category.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

ODS options

**URL=string-column**
specifies an HTML page that is displayed when a box or an outlier marker is selected.

Plot reference options

**NAME="string"**
assigns a name to this plot statement for reference in other template statements.

Statistics options

**DISPLAYSTATS=NONE | STANDARD | ALL | (statistics-list)**
specifies the statistics to be displayed for each box.
Required Arguments

Specifying only Y= creates a single box plot. Specifying both X= and Y= creates a box plot for each unique value of X.

**Y=** numeric-column | expression

specifies the column for the Y values. The Y values must be the statistical values needed for the box plot. At a minimum, there must be nonmissing values for the 25th and 75th percentiles.

**X=** column | expression

specifies the column for the X values. The X values must qualify or classify the values in the Y column. This optional argument is used to create a plot box for each classifier.

Restriction When you specify a numeric column for the X= argument and the category axis is an interval axis, you should not change the column value format. Doing so might result in incorrect output.

**STAT=** string-column

specifies the name of a column that contains the statistic represented by the value in the Y column. Valid STAT= column values include the following (see the requirements listed at the end of this description):

**BOXWIDTH**

specifies the width of the boxes as a ratio of the maximum possible width.

The range of values is 0 (narrowest) to 1 (widest). The default is 0.4. If the Y value corresponding to BOXWIDTH is in range, then it overrides the setting that is specified in the BOXWIDTH= option.

**DATAMAX**

specifies the maximum data value that includes not only the maximum whisker values but the maximum outlier values as well.

The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.

Note This statistic is valid in SAS 9.4M1 and later releases.

**DATAMIN**

specifies the minimum data value that includes not only the minimum whisker values but the minimum outlier values as well.

The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.

Note This statistic is valid in SAS 9.4M1 and later releases.

**FAROUTLIER**

specifies the observations that are outside the lower and upper far fences.

The far fences are located at a distance 3 times the Interquartile Range (IQR = Q3–Q1) above and below the box. The far outliers are labeled when the DATALABEL= option is used. Specify that LABELFAR= TRUE to label only the far outliers but not the outliers.

**MAX**

specifies the maximum data value less than or equal to the upper fence.

**MEAN**

specifies the data mean.
MEDIAN
specifies the data median.

MIN
specifies the minimum data value greater than or equal to the lower fence.

N
specifies the subgroup sample size.

The N value is not shown in the plot but is used to calculate notch locations when the DISPLAY= option displays notches.

OUTLIER
specifies the observations that are outside the lower and upper fences.

The fences are located at a distance 1.5 times the Interquartile Range (IQR = Q3–Q1) above and below the box. The outliers are labeled when the DATALABEL= option is used.

Q1
specifies the 1st quartile (25th percentile).

The data must contain a nonmissing value for this quartile.

Q3
specifies the 3rd quartile (75th percentile).

The data must contain a nonmissing value for this quartile.

STD
specifies the data standard deviation.

SUMWGT
specifies the sum of the weights for the box.

Note  This statistic is valid in SAS 9.4M1 and later releases.

Requirements
Nonmissing Y values for STAT observations of Q1 and Q3 are required in order for a box to be drawn. Other STAT values can be omitted or have missing Y values.

The STAT values, if present, must conform to the following rules in order for a box to be displayed:

\[
\begin{align*}
Q1 & \leq MEDIAN \leq Q3 \\
MIN & \leq MAX \\
STD & \geq 0 \\
N & > 0
\end{align*}
\]

Optional Arguments

BOXWIDTH=number
specifies the width of a box as a ratio of the maximum possible width.

Defaults
For a discrete category axis, 0.4 for a nongrouped box plot or 0.6 for a grouped box plot.

For an interval category axis, 85% of the smallest interval between any two boxes for the given plot.

Range 0–1, where 0 is the narrowest and 1 is the widest.
Interactions
For a grouped box plot with a discrete category axis when GROUPDISPLAY=CLUSTER, the box width is a percentage of the cluster width. See CLUSTERWIDTH.

This option is overridden by the Y value when the STAT= column value is BOXWIDTH and the corresponding Y value is within the range of 0–1.

Prior to SAS 9.4M3, for an interval box plot, this option is ignored, and the box width is controlled by the INTERVALBOXWIDTH= option. Starting with SAS 9.4M3, for an interval box plot, this option is ignored when the INTERVALBOXWIDTH= option is specified. Otherwise, the BOXWIDTH= option is honored.

CAPSCALE=positive-number
specifies a positive number to use as a multiplier for determining the default width of the whisker caps.

Note: This feature applies to SAS 9.4M5 and to later releases.

Default 1

Interaction This option is ignored when CAPSHAPE=NONE is in effect.

Tips Use the CAPSHAPE= option to specify the cap shape.

The thickness (height) of the SERIF, LINE, and BRACKET is determined by the thickness specified in the WHISKERATTRS= option. The cap line pattern is always solid.

Examples Double the size of the caps:
capscale=2

Reduce the size of the caps by 50%:
capscale=0.5

CAPSHAPE=SERIF | LINE | BRACKET | NONE
specifies the shape at the ends of the whiskers.

SERIF
specifies a short line perpendicular to the whisker.

LINE
specifies a line perpendicular to the whisker that extends the width of the box.

BRACKET
specifies a line perpendicular to the whiskers that extends the width of the box and that has short extensions at each end.

The extensions are drawn in the direction of the box.

NONE
specifies that no shape appears at the ends of the whiskers.

The following figure shows each of the shapes.
The cap color and the thickness are specified by the `WHISKERATTRS=` option. The cap pattern is always solid.

The `DISPLAY=` option must include `CAPS` in order for cap lines to be shown.

Starting with SAS 9.4M5, when `CAPSCALE=` is in effect, the display width for the caps is the cap's default width multiplied by the value specified in `CAPSCALE=`.

**CLUSTERWIDTH=number**

specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

| Default | 0.7 |
| Range   | 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width |
| Requirement | For this option to take effect, the `GROUP=` option must also be specified, and the `GROUPDISPLAY=` option must be set to `CLUSTER`. |

**CONNECT=MEAN | MEDIAN | Q1 | Q3 | MIN | MAX**

specifies that a connect line joins a statistic from box to box.

| Default | The GraphBox:Connect style reference. |
**Requirement**
The `DISPLAY=` option must contain the CONNECT `display-options` value in order for the connect line to be displayed.

**Interaction**
This option applies only when the `X=` argument is used to generate multiple boxes.

**Note**
Starting with SAS 9.4M3, the connect lines are drawn in axis order. In prior releases, they are drawn in data order.

**CONNECTATTRS=** `style-element | style-element (line-options) | (line-options)`
specifies the attributes of the lines connecting multiple boxes.

**Default**
The GraphConnectLine style element.

**Interactions**
- If there is only one box, then this option is ignored.
- If the `DISPLAY=` option does not include CONNECT, or if the `GROUP=` option is used, then this option is ignored.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Line Options” on page 1522 for available `line-options`.

**CONNECTBREAK=** `TRUE | FALSE`
specifies whether the connect line is broken for values that have no observations.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Default**
FALSE

**Requirement**
DISPLAY= must include CONNECT for this option to have any effect.

**Interaction**
This option is ignored when the `GROUP=` option is in effect.

**DATALABEL=** `column`
specifies the labels of the values that are identified as outlier or far outlier by the STAT= column. Either a numeric or a character column can be used.

**Default**
No data labels are displayed

**Interaction**
This option is ignored if `EXTREME= TRUE` or the `DISPLAY=` option does not display the outliers.

**See**
the `LABELFAR=` option

**DATALABELATTRS=** `style-element | style-element (text-options) | (text-options)`
specifies the color and font attributes of the outlier labels.

**Default**
The GraphDataText style element.

**Interaction**
This option is ignored if `EXTREME= TRUE` or the `DISPLAY=` option does not display the outliers.
See “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

See “Text Options” on page 1525 for available `text-options`.

**DATALABELSPLIT=TRUE | FALSE**

specifies whether to split the data labels at the specified split characters. When set to TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters.

**Default** FALSE. The data labels are not split.

**Requirement** The DATALABEL= option must also be specified.

**Interaction** The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

**See** “boolean" on page 1479 for other Boolean values that you can use.

**DATALABELSPLITCHAR="character-list"**

specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

```
"character-list"
```

one or more characters with no space between each character and enclosed in quotation marks.

**Default** A blank space

**Requirements** The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```
datalabelsplitchar="abc"
```

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction** The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes** When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip** Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.
**DATALABELSPLITCHARDROP**=TRUE | FALSE
specifies whether the split characters are included in the data labels.

TRUE
drops the split characters from the data label.

FALSE
includes the split characters in the data label.

When **DATALABELSPLIT**=TRUE and **DATALABELSPLITCHARDROP**=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:
- the data label text for this label is Product*Group*A
- **DATALABELSPLIT**=TRUE
- **DATALABELSPLITCHARDROP**=TRUE | FALSE
- **DATALABELSPLITCHAR**="*"

When **DATALABELSPLITCHARDROP**=TRUE, the asterisks are removed from the label. When **DATALABELSPLITCHARDROP**=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default TRUE. The split characters are dropped from the data label.

Requirement The **DATALABEL**= option and the **DATALABELSPLIT**=TRUE option must also be specified.

Interaction The **DATALABELSPLITCHAR**= option specifies the split characters.

See "boolean " on page 1479 for other Boolean values that you can use.

**DATALABELSPLITJUSTIFY**=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the data label blocks.

AUTO
justifies the labels based on the **DATALABELPOSITION**= option.

<table>
<thead>
<tr>
<th><strong>DATALABELPOSITION</strong>= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
</tbody>
</table>
### DATALABELPOSITION= Value | Justification

<table>
<thead>
<tr>
<th>Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

**CENTER**

**LEFT**

**RIGHT**

justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.

**Note:** The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.

In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

**Default**

AUTO

**Requirement**

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction**

This option has no effect if DATALABELPOSITION=AUTO.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**

enhances the visual appearance of the filled boxes. The following figure shows boxes with each of the skins applied.
The **DATASKIN**= option value that is specified in the **BEGINGRAPH** statement. If that value is not specified, then the `GraphSkins:DataSkin` style element value is used.

**Restriction**
Starting with **SAS 9.4M1**, the maximum number of skinned graphical elements is limited to 200 per plot in an **OVERLAY** or **PROTOTYPE** layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the **DATASKINMAX**= option in your ODS GRAPHICS statement to increase the maximum limit.

**Requirement**
For this option to have any effect, the fill must be enabled by the ODS style or the **DISPLAY**= option.

**Interactions**
This option overrides the **BEGINGRAPH** statement **DATASKIN**= option.

The data skin appearance is based on the **FILLATTRS**= color.

When a data skin is applied, all bar outlines are set by the skin, and the **OUTLINEATTRS**= option is ignored.

**DATATRANSPARENCY**= **number**
specifies the degree of the transparency of the box outlines, box fill, whiskers, mean, median, caps, connect lines, outliers, and data labels, if displayed.

**Default**
0

**Range**
0–1, where 0 is opaque and 1 is entirely transparent

**Tip**
The **FILLATTRS**= option can be used to set transparency for just the interior fill area of the boxes. You can combine this option with **FILLATTRS**= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

datatransparency=0.2 fillattrs=(transparency=0.6)
**DISCRETEOFFSET=**number  
specifies an amount to offset all boxes from the discrete X ticks.

- **Default**: 0 (no offset, all boxes are centered on the discrete ticks)
- **Range**: -0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is to the right when ORIENT=VERTICAL, and up when ORIENT=HORIZONTAL. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)
- **Restriction**: This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

**Tip**: Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**See**: “About the DISCRETEOFFSET= Option” on page 409

**DISPLAY=**(display-options) | STANDARD | ALL  
specifies which box-plot features to display.

- **display-options**: specifies a space-separated list of one or more display options, enclosed in parentheses.
- The list can contain one or more of the following options:
  - **CAPS**: displays caps at the ends of the whiskers
  - **CONNECT**: displays the line connecting multiple boxes
  - **FILL**: displays the box fill color
  - **FILLPATTERN**: displays the box fill pattern

**Note**: This feature applies to SAS 9.4M5 and to later releases.

**Tip**: By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the FILLPATTERNATTRS= option to specify a fill pattern for the boxes.

- **MEAN**: displays the mean symbol within the box
- **MEDIAN**: displays the median line within the box
- **NOTCHES**: displays notched boxes
- **OUTLIERS**: displays markers for the outliers
STANDARD
specifies CAPS FILL MEAN MEDIAN, and OUTLIERS

ALL
specifies all features: CAPS, CONNECT, FILL, FILLPATTERN, MEAN, MEDIAN, NOTCHES, and OUTLIERS

The endpoints of the notches are at the following computed locations:

\[
\text{median} \pm \frac{1.58 \cdot \text{IQR}}{\sqrt{n}}
\]

In the equation, IQR is the interquartile range and \( n \) is the sample size.

---

**Default**
The GraphBox:DisplayOpt style reference. If this style element does not exist, then the default is STANDARD.

**Restriction**
The display features requested can be displayed only if the input data includes this information.

**Interaction**
If **EXTREME**= TRUE, then the OUTLIERS feature is ignored.

**Notes**
Starting with SAS 9.4M3, the connect lines are drawn in axis order. They are drawn in data order in prior releases.

Starting with SAS 9.4M3, when **DISPLAY**= includes MEAN, the BOXPLOTPARM statement contributes its mean markers to a discrete legend when **TYPE**=MARKER is in effect in the DISCRETELEGEND statement.

**Tips**
To control the appearance of these features, use the

**CONNECTATTRS**, **FILLATTRS**, **MEANATTRS**, **MEDIANATTRS**, **OUTLIERATTRS**, and **WHISKERATTRS** options. The **WHISKERATTRS** option controls affects both CAPS and WHISKERS. Starting with SAS 9.4M5, use the **FILLPATTERNATTRS** on page 394 option to control the appearance of the box fill pattern.
Regardless of which display options are being displayed, this option does not affect the axis range.

**DISPLAYSTATS=NONE | STANDARD | ALL | (statistics-list)**

specifies the statistics to be displayed for each box.

**NONE**

does not display any statistics.

**STANDARD**

displays N, MEAN, and STD.

**ALL**

displays all available statistics.

**(statistics-list)**

specifies a space-separated list of one or more statistics, enclosed in parentheses.

The list can include one or more of the following:

**DATAMAX**

maximum data value that includes not only the maximum whisker values but also the maximum outlier values.

This option is valid in *SAS 9.4M1* and later releases. The DATAMAX value is greater than or equal to the MAX value and is always represented in the axis range.

**DATAMIN**

minimum data value that includes not only the minimum whisker values but also the minimum outlier values.

This option is valid in *SAS 9.4M1* and later releases. The DATAMIN value is less than or equal to the MIN value and is always represented in the axis range.

**IQR**

interquartile range (Q3–Q1).

**MAX**

maximum data value below the box upper fence.

**MEAN**

mean data value for the box.

**MEDIAN**

median data value for the box.

**MIN**

minimum data value above the box lower fence.

**N**

number of observations for the box.

**Q1**

lower quartile (25th percentile) for the box.

**Q3**

upper quartile (75th percentile) for the box.

**RANGE**

range of the data (MAX–MIN).

**STD**

standard deviation of the data for the box.
**SUMWGT**

sum of the weights for the box.

This option is valid in **SAS 9.4M1** and later releases.

**Default**

NONE

**Restrictions**

Prior to **SAS 9.4M5**, this option is ignored if **ORIENT=HORIZONTAL**. Starting with **SAS 9.4M5**, statistics can be displayed for both vertical and horizontal box plots.

This option is ignored if the **GROUP=** option is also specified.

Only those statistics that are included in the **STAT=** column can be displayed. **RANGE** requires both **MAX** and **MIN** to be included. **IQR** requires both **Q1** and **Q3** to be included.

**Note**

The notches in the box plot can extend beyond **DATAMIN** and **DATAMAX** in some cases.

**EXTREME=TRUE | FALSE**

specifies whether the whiskers can extend beyond the fences. Fences are locations above and below the box. The upper and lower fences are located at a distance 1.5 times the Interquartile Range (IQR) \((IQR = Q3 - Q1)\). The upper and lower far fences are located at a distance 3 times the IQR.

**FALSE**

specifies that whiskers be drawn from the upper edge of the box to the largest value within the upper fence, and from the lower edge of the box to the smallest value within the lower fence.

This representation is sometime called a schematic box plot or a Tukey box plot.

**TRUE**

specifies that whiskers be drawn to the largest and smallest data values, whether these values are inside or outside the fences.

The outliers and far outliers are not displayed and are not labeled. This representation is sometime called a **skeletal box and whisker plot**.

**Default**

FALSE

**Interaction**

This option overrides the **DATALABEL=**, **DATALABELATTRS=**, **LABELFAR=**, **OUTLIERATTRS=**, and **SPREAD=** options.

**See**

“**Statement Summary**” on page 406

“**boolean**” on page 1479 for other Boolean values that you can use.

**FILLATTRS=** *style-element | style-element (fill-options) | (fill-options)*

specifies the appearance of the interior fill area of the boxes.

**Defaults**

For non-grouped data, the Color attribute of GraphDataDefault style element.

For grouped data, the Color attribute of GraphData1–GraphDataN style elements.
Interaction For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

Tip The DATATRANSPARENCY= option sets the transparency for the box outlines, box fill, whiskers, mean, median, caps, connect lines, and outliers. You can combine this option with DATATRANSPARENCY= to set one transparency for the box outlines and the whiskers, mean, median, caps, and connect lines, but a different transparency for the box fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

**FILLPATTERNATTRS=**

|style-element| (fill-pattern-options)

specifies the appearance of the pattern-filled areas.

Note: This feature applies to SAS 9.4M5 and to later releases.

Defaults For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

Interaction For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

Tip In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Pattern Options” on page 1521 for available fill-pattern-options.

**GROUP=**

column | discrete-attr-var | expression

creates a box plot for each unique group value of the specified column.

discrete-attr-var

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

The box plot display depends on the setting for the GROUPDISPLAY= option. This option can be used to group the box plots in the display.
A distinct box is created for each group value by varying the visual attributes of the box display features. The display features are controlled by the current ODS style or by the \texttt{DISPLAY=} option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature ¹</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color and \texttt{ContrastColor} attributes of a \texttt{GraphData1–GraphDataN style element} or the \texttt{GraphMissing style element}. The \texttt{Color} attribute controls the fill color, and the \texttt{ContrastColor} attribute controls the outline color. The box outline line style is solid.</td>
</tr>
<tr>
<td>Fill pattern ²</td>
<td>\texttt{FillPattern} and \texttt{ContrastColor} attributes of a \texttt{GraphData1–GraphDataN style element} or the \texttt{GraphMissing style element}. The \texttt{ContrastColor} attribute controls the color of the fill pattern and box outline. The box outline line style is solid. If the active ODS style does not specify \texttt{FillPattern}, fill patterns are not displayed by default.</td>
</tr>
<tr>
<td>Median, caps, and notches</td>
<td>\texttt{ContrastColor} attribute of a \texttt{GraphData1–GraphDataN style element} or the \texttt{GraphMissing style element} with a solid line style.</td>
</tr>
<tr>
<td>Mean and outliers</td>
<td>\texttt{MarkerSymbol} and \texttt{ContrastColor} attributes of a \texttt{GraphData1–GraphDataN style element} or the \texttt{GraphMissing style element}.</td>
</tr>
</tbody>
</table>

¹ The default display features are determined by the DisplayOpt attribute of the GraphBox style element.
² Fill patterns are valid for box plots starting with \texttt{SAS 9.4M5}.

**Defaults**

If box fills are enabled by the ODS style or by the \texttt{DISPLAY=} option, then each distinct group value is represented in the plot by a different fill color. Starting with \texttt{SAS 9.4M5}, each distinct group value can be represented by a different fill pattern, with or without a different fill color. The fill color and fill pattern are defined by the \texttt{Color} and \texttt{FillPattern} attributes of the \texttt{GraphData1–GraphDataN} and \texttt{GraphMissing style elements}.

If only box outlines are enabled by the ODS style or by the \texttt{DISPLAY=} option, then each distinct group value is represented in the plot by a different box outline color and line style. The outline colors and line styles are defined by the \texttt{ContrastColor} and \texttt{LineStyle} attributes of the \texttt{GraphData1–GraphDataN} style elements.

Each distinct group value is represented in the plot by a different box outline color. The outline colors are defined by the \texttt{ContrastColor} attribute of the \texttt{GraphData1–GraphDataN} and \texttt{GraphMissing style elements}.

If the \texttt{MEDIAN}, \texttt{CAPS}, or \texttt{NOTCHES} are enabled by the \texttt{DISPLAY=} option, then each of these features uses the same color as the box outline. Line styles do not change by group value.

If the \texttt{MEAN} or \texttt{OUTLIERS} are enabled by the \texttt{DISPLAY=} option, then each distinct group value is represented by a different marker.
The markers are defined by the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN and GraphMissing style elements. A marker is used for both MEAN and OUTLIERS, if displayed.

**Interactions**

Connect lines are not drawn for grouped data.

This option causes the DISPLAY=(CONNECT) and the CONNECT= options to be ignored.

**Tips**

By default, the group values are mapped in the order of the data. The GROUPORDER= option can be used to control the sorting order of the group values. The INDEX= option can be used to alter the default sequence of colors and markers.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the box outlines, but the PATTERN= setting on the OUTLIERATTRS= option could be used to assign the same line pattern to all box outlines and connect lines.

**See**

the GROUPDISPLAY= option to see the output for the grouped boxes

"DISCRETEATTRVAR Statement" on page 1435

**GROUPDISPLAY=OVERLAY | CLUSTER**

specifies how to display the boxes that represent group values for the coordinate pairs. The following example shows a box plot with GROUPDISPLAY=CLUSTER:

---

**City Mileage for Vehicle Types**

- **OVERLAY**
  - draws boxes for a given group value at the exact coordinate.
  - Depending on the data, boxes at a given coordinate might overlap.
CLUSTER
draws boxes for a given group value adjacent to each other.
This option is available only when the category (X) column is discrete.

Default
OVERLAY

Interactions
This option is ignored unless GROUP= is specified.

The groups in the legend are shown in the order that is specified in
GROUPORDER by default.

Attributes such as color, symbol, and pattern are assigned to each
group in DATA order by default.

Tip
Use the CLUSTERWIDTH= option to control the width of the
clusters when CLUSTER is in effect.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in the group-column data order.

REVERSEDATA
orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip
This option is useful when you want to reverse the category axis.

ASCENDING
orders the groups within a category in ascending order.

DESCENDING
orders the groups within a category in descending order.

Defaults
DATA if the data is a SAS data set.
ASCENDING if the data is a CAS in-memory table. Sorting is based
on nonformatted values for numeric group values or on formatted
values for character group values.

Interactions
This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is
specified in GROUPORDER.

Notes
When plotting a SAS data set, attributes such as color, symbol, and
pattern are assigned to each group in DATA order by default. When
plotting a CAS in-memory table, attributes are assigned in
ascending order of the group column character values or
unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the
group values within each category (or X value) for display position
purposes only. For numeric data, the order is based on the
unformatted values. For character data, the order is based on the
formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In **SAS 9.4M6** and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

**Default** TRUE

**Interaction** For this option to take effect, the GROUP= option must also be specified.

**Tip** The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See** “boolean ” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**
specifies indices for mapping box attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**Requirements** The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction** For this option to take effect, the GROUP= option must also be specified.

**Notes** The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip** You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.
INTERVALBOXWIDTH=AUTO | *dimen*

specifies the box width when an interval category (X) column is specified.

**AUTO**

the system determines the box width.

uses one of the following box-width values listed in the order of precedence:

- the BOXWIDTH statistic, if it is included in the Stat column and if the Y value is within the range of 0–1
- starting with *SAS 9.4M3*, the BOXWIDTH= option value, if it is specified
- otherwise, 85% of the smallest interval between any two boxes for the given plot

**dimen**

sets the box width to the specified value.

**Default**

AUTO

**Requirement**
The category axis (X by default) must be an interval axis for the INTERVALBOXWIDTH= option to have any effect.

**Interactions**

Starting with *SAS 9.4M3*, this option is overridden by the Y value when the STAT= column value is BOXWIDTH and the corresponding Y value is within the range of 0–1.

Prior to *SAS 9.4M3*, this option controls the box width for an interval box plot. Starting with *SAS 9.4M3*, this option overrides the BOXWIDTH= option for an interval box plot.

**Tip**
The box width that you specify with this option is honored even if the boxes overlap. If the boxes overlap, reduce the interval box width or use the BOXWIDTH= option instead.

**See**

“*dimension*” on page 1480

LABELFAR=TRUE | FALSE

specifies whether all outliers or only far outliers are labeled.

**FALSE**

applies the labels specified by the DATALABEL= option to both the outliers and the far outliers.

**TRUE**

applies the labels specified by the DATALABEL= option to the far outliers.

**Default**

FALSE

**Interaction**
This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

**See**

“*DATALABEL=column* ” on page 385

“Statement Summary” on page 366 for information about outliers.

“*boolean* ” on page 1479 for other Boolean values that you can use.

**LEGENDLABEL=“*string*”**

specifies a label to be used in a discrete legend for this plot.
The Y= column label. If a label is not defined, then the Y= column name is used.

This option applies only to an associated DISCRETELEGEND statement.

If the GROUP= option is specified, then this option is ignored.

**MEANATTRS**=

specifies the attributes of the marker representing the mean within the box.

For non-grouped data, GraphBoxMean style element.

For grouped data, the MarkerSymbol, Markersize, and ContrastColor attributes of the GraphData1–GraphDataN style elements.

This option is ignored if the DISPLAY= option does not display the mean.

**MEDIANATTRS**=

specifies the appearance of the line representing the median within the box.

For non-grouped data, the GraphBoxMedian style element.

For grouped data, the LineStyle and LineThickness attributes of the GraphBoxMedian style element, and the ContrastColor attribute of the GraphData1–GraphDataN style elements.

This option is ignored if the DISPLAY= option does not display the median.

**NAME**="string"

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

The string is used as the default legend label if the LEGENDLABEL= option is not used.

**ORIENT**=VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and of the boxes.
OUTLIERTIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over an outlier. If this option is used, then it replaces all of the information that is displayed by default.

(role-list)
an ordered, space-separated list of unique BOXPLOTPARM roles and user-defined roles. BOXPLOTPARM roles for OUTLIERTIP include X, Y, STAT, and DATALABEL.

Note
In the data tip, the STAT role displays the text “outlier” or “far outlier” as applicable.

Tip
User-defined roles are defined with the ROLENAME= option.

Example
The following example displays data tips for the columns that are assigned to the X and Y roles, and also the data column Obs, which is not assigned to any pre-defined BOXPLOTPARM role. The Obs column must first be assigned a role:

ROLENAME=(TIP1=OBS)
OUTLIERTIP=(X Y TIP1)

Default
The columns assigned to these roles are automatically included in the data tip information: X and Y.

Requirement
To generate data tips, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and write the graphs to the ODS HTML destination.

Interaction
The labels and formats for the OUTLIERTIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

See
the ROLENAME= option for specifying user-defined roles.
the TIP= option for specifying the information to display when the cursor is positioned over a box or whisker in the box plot.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the box outline.
Defaults
For non-grouped data, the ContrastColor, LineThickness, andLineStyle attributes of the GraphOutlines style element.

For grouped data and filled boxes, theLineStyle and LineThickness attributes of the GraphOutlines style element, and the ContrastColor attribute of the GraphData1–GraphDataN style elements.

For grouped data and unfilled boxes, the LineThickness attribute of the GraphOutlines style element, and the ContrastColor andLineStyle attributes of the GraphData1–GraphDataN style elements.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDDED or LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

ROLENAMEN=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the OUTLIERTIP= option.

(role-name-list)
a space-separated list of role-name = column pairs.

Example The following example assigns the column Obs to the user-defined role TIP:

ROLENAME=(TIP1=OBS)

Default No user-defined roles

Requirement The role names that you choose must be unique and different from the predefined roles DATAMAX, DATAMIN, MAX, MIN, MEAN, MEDIAN, N, Q1, Q2, STD, SUMWGT, and X.

Note The roles DATAMAX, DATAMIN, and SUMWGT are valid in SAS 9.4M1 and in later releases.
SPREAD=TRUE | FALSE
specifies whether outliers with the same value are spread out to avoid overlap. For vertical box plots this means offsetting the outliers horizontally. If this option is false, then outliers with the same value are plotted in the same position, which means only one is visible.

Default FALSE

Interaction This option is ignored if EXTREME= TRUE or the DISPLAY= option does not display the outliers.

See “boolean ” on page 1479 for other Boolean values that you can use.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a box or whisker in the box plot. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the box plot can be specified along with roles that do contribute.

(role-list)
an ordered, space-separated list of unique BOXPLOTPARM roles.

BOXPLOTPARM roles for TIP include DATAMAX, DATAMIN, MAX, MIN, MEAN, MEDIAN, N, Q1, Q2, STD, SUMWGT, and X.

Note The roles DATAMAX, DATAMIN, and SUMWGT are valid in SAS 9.4M1 and later releases.

Tip Statistics such as N, MIN, and MAX are special roles, They are not column-based like the X role.

Example The following example displays data tips only for the columns that are assigned to the roles X (CATEGORY) and the statistic MEAN:

TIP=(X MEAN)

NONE
suppresses data tips from the plot.

Default The columns assigned to these roles are automatically included in the data tip information: DATAMAX, DATAMIN, MAX, MIN, MEAN, MEDIAN, N, Q1, Q2, STD, SUMWGT, and X.

Requirement To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

See the OUTLIERTIP= option for specifying the information to display when the cursor is positioned over an outlier.
**TIPFORMAT=(role-format-list)**
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

*role-format-list*
a space-separated list of role-name = format pairs.

**Example**
```
ROLENAME=(TIP1=SAI)  
TIP=(TIP1)  
TIPFORMAT=(TIP1=DOLLAR12.)
```

**Default**
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**
Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

*role-label-list*
a space-separated list of rolename ="string" pairs.

**Example**
```
ROLENAME=(TIP1=PCT)  
TIP=(TIP1)  
TIPLABEL=(TIP1="Percent")
```

**Default**
The column label or column name of the column assigned to the role.

**Restriction**
Only the roles that appear in the OUTLIERTIP= or TIP= options are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**URL=string-column**
specifies an HTML page that is displayed when a box or an outlier marker is selected.

---

**Note:** This option is valid starting in SAS 9.4M1.

---

*string-column*
specifies a column that contains a valid HTML page reference (HREF) for each box or outlier marker that is to have an active link.

**Example**
```
http://www.sas.com
```

The character column can specify different target URLs for each box and outlier marker.

**Requirements**
The target URLs for the boxes must be specified in the Q1 statistic observations, and the target URLs for the outlier markers
must be specified in the OUTLIER statistic observations. URLs that are specified in observations other than Q1 and OUTLIER are ignored.

To generate selectable boxes and outlier markers, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tips

The URL value can be blank for a box or outlier, meaning that no action is taken when that box or outlier marker is selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

Example

The following vehicle mileage data sample shows box and outlier URLs specified in column URL for Sedan.

<table>
<thead>
<tr>
<th>STAT</th>
<th>X</th>
<th>VALUE</th>
<th>DATALABEL</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Sedan</td>
<td>262.0</td>
<td></td>
<td>./mileageSedan.html</td>
</tr>
<tr>
<td>MEAN</td>
<td>Sedan</td>
<td>08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIAN</td>
<td>Sedan</td>
<td>20.00</td>
<td></td>
<td>./mileageSedan.html</td>
</tr>
<tr>
<td>Q1</td>
<td>Sedan</td>
<td>18.00</td>
<td></td>
<td>./mileageSedan.html</td>
</tr>
<tr>
<td>Q3</td>
<td>Sedan</td>
<td>24.00</td>
<td></td>
<td>./mileageSedan.html</td>
</tr>
<tr>
<td>STD</td>
<td>Sedan</td>
<td>4.23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OUTLIER</td>
<td>Sedan</td>
<td>36.00</td>
<td>Honda</td>
<td>./mileageHonda.html</td>
</tr>
<tr>
<td>OUTLIER</td>
<td>Sedan</td>
<td>35.00</td>
<td>Toyota</td>
<td>./mileageToyota.html</td>
</tr>
<tr>
<td>MIN</td>
<td>Sedan</td>
<td>12.00</td>
<td>Volvo</td>
<td></td>
</tr>
<tr>
<td>MAX</td>
<td>Sedan</td>
<td>33.00</td>
<td>Volvo</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**WHISKERATTRS=style-element | style-element (line-options) | (line-options)**

specifies the line properties of the whiskers and caps.

Defaults

For non-grouped data, the GraphBoxWhisker style element.

For grouped data, the LineStyle and LineThickness attributes of the GraphBoxWhisker style element, and the ContrastColor attribute of the GraphData1–GraphDataN style elements.

Restriction

The caps are always drawn with a solid line.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.
**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

<table>
<thead>
<tr>
<th>Default</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>This option is ignored if the X= argument is not specified.</td>
</tr>
</tbody>
</table>

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

<table>
<thead>
<tr>
<th>Default</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.</td>
</tr>
</tbody>
</table>

**Details**

**Statement Summary**

The BOXPLOTPARM statement requires precomputed input data. One reason to choose this statement over the BOXPLOT statement is that you can control the computational technique used to compute various statistics for the box plot, such as the mean, quartiles, location of fences, outlier definition, and so on. See Appendix 6, “Generalized Macro for BOXPLOTPARM Data,” on page 1545 for examples of such computations using PROC SUMMARY and multiple DATA steps.

The BOXPLOTPARM statement displays a single box if given just Y and a STAT argument. It displays multiple boxes if given both Y and X and a STAT argument and X has more than one unique value.

By default for numeric or character columns, the category (X) axis is TYPE=DISCRETE. You can override the default and specify TYPE=LINEAR in the parent layout, provided that the X column is numeric. The axis for the analysis (Y) column is always LINEAR. When the X axis is LINEAR, you can use the INTERVALBOXWIDTH= option to specify the box width.

When ORIENT= VERTICAL, the X (or X2) axis is used for the X column and the Y (or Y2) axis is used for the Y column. When ORIENT=HORIZONTAL, the X (or X2) axis is used for the Y column and the Y (or Y2) axis is used for the X column.

Two basic box plot representations can be drawn with the BOXPLOTPARM statement: a schematic (Tukey) box plot and a skeletal box plot. See the EXTREME= option for details.

The following figure illustrates the box plot elements:
As shown in the figure, the bottom and top edges of the box are located at the 1st quartile (25th percentile) and 3rd quartile (75th percentile) of the sample. Within the box, you can display the median (50th percentile) as a line and the mean as a marker (see the DISPLAY= option).

You can also display markers and data labels for outliers. Outliers are observations that are more extreme than the upper and lower fences ($\pm 1.5 \text{IQR}$). Outliers that are beyond upper and lower far fences ($\pm3 \text{IQR}$) are called FAR OUTLIERS and can also be identified and labeled. From a graphical perspective, the locations of fences along the axis are known, but there is no line or marker that displays a fence. (See DISPLAY=, LABELFAR=, and DATALABEL= options).

Finally, you can control the range represented by the whiskers. By default, the whiskers are drawn from the upper edge of the box to the MAX value, and from the lower edge of the box to the MIN value. (See the EXTREME= option.)

Input Data Requirements for the BOXPLOTPARM Statement

At a minimum, valid data for the BOXPLOTPARM statement must provide a numeric column (Y=) that contains calculated statistics for an analysis, and a string column (STAT=) that identifies each statistic. The Y column must contain nonmissing values for the Q1 (25th percentile) and Q3 (75th percentile) statistics. If Y values are missing or not supplied for other statistic values, then those statistics are not displayed in the plot, regardless of syntax requests to display them.

For example, a petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil more viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following data show the statistics that are calculated for one day of this process:
To plot the data from the preceding table, the following `BOXPLOTPARM` statement uses the `Y=` and `STAT=` arguments to generate a single box plot for the recorded statistics:

```
BOXPLOTPARM Y=PowerOutputs STAT=Statistic;
```

If the data contain statistics for multiple days of the process, then a third column in the data must be present to identify the days that the statistics were recorded. For example, the following data show the statistics that are calculated for two days of this process:

<table>
<thead>
<tr>
<th>Day</th>
<th>PowerOutputs</th>
<th>Statistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>04JUL</td>
<td>3180.00</td>
<td>MIN</td>
</tr>
<tr>
<td>04JUL</td>
<td>3340.00</td>
<td>Q1</td>
</tr>
<tr>
<td>04JUL</td>
<td>3487.40</td>
<td>MEAN</td>
</tr>
<tr>
<td>04JUL</td>
<td>3490.00</td>
<td>MEDIAN</td>
</tr>
<tr>
<td>04JUL</td>
<td>3610.00</td>
<td>Q3</td>
</tr>
<tr>
<td>04JUL</td>
<td>4050.00</td>
<td>MAX</td>
</tr>
<tr>
<td>04JUL</td>
<td>20.00</td>
<td>N</td>
</tr>
<tr>
<td>Day</td>
<td>PowerOutputs</td>
<td>Statistic</td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>04JUL</td>
<td>3610.00</td>
<td>Q3</td>
</tr>
<tr>
<td>04JUL</td>
<td>4050.00</td>
<td>MAX</td>
</tr>
<tr>
<td>04JUL</td>
<td>20.00</td>
<td>N</td>
</tr>
<tr>
<td>05JUL</td>
<td>3179.00</td>
<td>MIN</td>
</tr>
<tr>
<td>05JUL</td>
<td>3333.50</td>
<td>Q1</td>
</tr>
<tr>
<td>05JUL</td>
<td>3471.65</td>
<td>MEAN</td>
</tr>
<tr>
<td>05JUL</td>
<td>3419.50</td>
<td>MEDIAN</td>
</tr>
<tr>
<td>05JUL</td>
<td>3605.00</td>
<td>Q3</td>
</tr>
<tr>
<td>05JUL</td>
<td>3849.00</td>
<td>MAX</td>
</tr>
<tr>
<td>05JUL</td>
<td>20.00</td>
<td>N</td>
</tr>
</tbody>
</table>

To plot the data from the preceding table, the BOXPLOTPARM statement needs the `Y=`, `STAT=`, and `X=` arguments to generate a separate box plot for each day that the statistics were recorded:

```plaintext
BOXPLOTPARM Y=PowerOutputs STAT=Statistic X=Day;
```

See Appendix 6, “Generalized Macro for BOXPLOTPARM Data,” on page 1545 for a more complete example of providing input data for BOXPLOTPARM.

About the DISCRETEOFFSET= Option

The `DISCRETEOFFSET=` option is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple BOXPLOTPARM statements are used with different analysis variables, then the boxes for matching X values are centered on the ticks. Depending on the data, the
boxes might be superimposed. The following code fragment shows the default box positioning:

```sas
layout overlay / cycleattrs=true
   yaxisopts=(label="Miles Per Gallon");

   boxplotparm x=type y=mpg_city stat=y_stat / name="City";
   boxplotparm x=type y=mpg_highway stat=y_stat / name="Highway";

   discretelegend "City" "Highway";
endlayout;
```

To place the different response values side by side, you can assign a different offset to each BOXPLOTPARM statement. The BOXWIDTH= option can be used in conjunction with the DISCRETEOFFSET= option to create narrower boxes when desired.

```sas
layout overlay / cycleattrs=true
   yaxisopts=(label="Miles Per Gallon");

   boxplotparm x=type y=mpg_city stat=y_stat / name="City"
       discreteoffset=0.2;
   boxplotparm x=type y=mpg_highway stat=y_stat / name="Highway"
       discreteoffset=-0.2;

   discretelegend "City" "Highway";
endlayout;
```
Changing Box Plot Display

SAS defines graphical style elements that control the display of box plots. Using these style elements as a starting point, you can change the style attribute values to achieve a very different appearance for your box plots. For more information, see “Changing Box Plot Display” on page 369.

Example: BOXPLOT PARM Statement

The following graph was generated by the “Example Program” on page 412:
Use the %BOXCOMPUTE macro to generate the data for this example. See Appendix 6, "Generalized Macro for BOXPLOTPARM Data," on page 1545 to see the code for creating all of the data.

%boxcompute(indsn=sashelp.cars,x=type,y=mpg_city,datalabel=make);

### Example Program

Here is the code for this example.

```sas
/* Define the template for the plot. */
proc template;
   define statgraph boxplotparm1;
      begingraph;
         entrytitle "City Mileage for Vehicle Types";
         layout overlay;
            boxplotparm y=value x=x stat=stat /
               datalabel=datalabel spread=true ;
         endlayout;
      endgraph;
   end;
run;

/* Generate the plot. */
proc sgrender data=boxdata template=boxplotparm1;
run;
```

### BUBBLEPLOT Statement

Creates a bubble plot of the input data. The locations of the bubble centers correspond to the values of X and Y columns in the data, and the bubble radii correspond to the values of a SIZE column.

#### Tip

Starting with **SAS 9.4M3**, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify `SUBPIXEL=OFF` in the `BEGINGRAPH` statement or in an ODS GRAPHICS statement. For information about the `BEGINGRAPH` statement `SUBPIXEL=` option, see `SUBPIXEL=` on page 35. For information about the ODS GRAPHICS statement `SUBPIXEL=` option, see "ODS GRAPHICS Statement" on page 1483.

#### Syntax

```sas
BUBBLEPLOT X=column | expression
     Y=column | expression
     SIZE=numeric-column | expression <options>;
```

#### Summary of Optional Arguments

##### Appearance options

- `BUBBLERADIUSMAX=dimension`
specifies the drawing size of the largest bubble.

**BUBBLERADIUSMIN=** *dimension*
specifies the drawing size of the smallest bubble.

**COLORMODEL=** *color-ramp-style-element | (color-list)*
specifies a color ramp to use with the COLORRESPONSE= option.

**COLORRESPONSE=** *numeric-column | range-attr-var | expression*
specifies the numeric column or range attribute map variable to use to determine the bubble colors.

**DATASKIN=** *NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN*
enhances the visual appearance of the filled bubbles.

**DATATRANSPARENCY=** *number*
specifies the degree of the transparency of the bubble fills and bubble outlines

**DISPLAY=** *(display-options) | STANDARD | ALL*
specifies which bubble features to display.

**DRAWORDER=** *SIZE | DATA*
specifies whether the bubbles are drawn according to bubble size or according to data order.

**FILLATTRS=** *style-element | style-element (fill-options) | (fill-options)*
specifies the appearance of the filled bubble areas.

**FILLPATTERNATTRS=** *style-element | (fill-pattern-options)*
specifies the appearance of the pattern-filled areas.

**INDEX=** *positive-integer-column | expression*
specifies indices for mapping bubble attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**OUTLINEATTRS=** *style-element | style-element (line-options) | (line-options)*
specifies the appearance of the bubble outlines.

**RELATIVESCALE=** *TRUE | FALSE*
specifies whether the SIZE= column values are interpreted as relative values.

**RELATIVESCALETYPE=** *LINEAR | PROPORTIONAL*
specifies the type of scaling that is to be applied to the SIZE= column values.

**REVERSECOLORMODEL=** *TRUE | FALSE*
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

**SIZE_THRESHOLD_MAX=** *numeric-value*
specifies a SIZE= column value threshold at which bubble size is clamped to the BUBBLERADIUS_MAX= option value.

### Axes options

**PRIMARY=** *TRUE | FALSE*
specifies that the data columns for this plot and the plot type be used for determining default axis features.

**XAXIS=** *X | X2*
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS=** *Y | Y2*
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

### Data tip options
ROLENAME=(role-name-list)
specifies user-defined roles that can be used to display information in the
data tips.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over the
bubbles.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns.

TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options
DATALABEL=column | expression
specifies a column for bubble labels.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the bubble labels.

DATALABELPOSITION=TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the bubble labels relative to the bubble.

DATALABELSPLIT=TRUE | FALSE
specifies whether to split the data labels at the specified split characters.

DATALABELSPLITCHAR="character-list"
specifies one or more characters on which the data labels can be split if
needed.

DATALABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the data labels.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the data label
blocks.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options
GROUP=column | discrete-attr-var | expression
creates a separate bubble color for each unique grouping that is specified.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the
plot.

ODS options
URL=string-column
specifies an HTML page to display when a bubble is selected.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template
statements.

Required Arguments
X=column | expression
specifies the column for the X values of the bubble centers.
**Y=column | expression**  
specifies the column for the Y values of the bubble centers.

**SIZE=numeric-column | expression**  
specifies the bubble SIZE values.

**Optional Arguments**

**BUBBLERADIUSMAX=dimension**  
specifies the drawing size of the largest bubble.

<table>
<thead>
<tr>
<th>Default</th>
<th>Three times as large as the size set by GraphDataDefault:markerSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>The <em>dimension</em> value must be greater than the BUBBLERADIUSMIN= <em>dimension</em> value.</td>
</tr>
<tr>
<td>Interaction</td>
<td>This option is ignored when RELATIVESCALE= FALSE.</td>
</tr>
<tr>
<td>Tip</td>
<td>A maximum size that is specified as a percent is interpreted as a percent of the graph's height. The height can be adjusted with the DESIGNHEIGHT= option in the BEGINGRAPH statement or the HEIGHT= option in the ODS GRAPHICS statement.</td>
</tr>
<tr>
<td>See</td>
<td>“dimension” on page 1480</td>
</tr>
</tbody>
</table>

**BUBBLERADIUSMIN=dimension**  
specifies the drawing size of the smallest bubble.

<table>
<thead>
<tr>
<th>Default</th>
<th>GraphDataDefault:markerSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>The BUBBLERADIUSMIN= value must be less than the BUBBLERADIUSMAX= value.</td>
</tr>
<tr>
<td>Interaction</td>
<td>This option is ignored when RELATIVESCALE= FALSE.</td>
</tr>
<tr>
<td>Tip</td>
<td>A maximum size that is specified as a percent is interpreted as a percent of the graph's height. The height can be adjusted with the DESIGNHEIGHT= option in the BEGINGRAPH statement or the HEIGHT= option in the ODS GRAPHICS statement.</td>
</tr>
<tr>
<td>See</td>
<td>“dimension” on page 1480</td>
</tr>
</tbody>
</table>

**COLORMODEL=color-ramp-style-element | (color-list)**  
specifies a color ramp to use with the COLORRESPONSE= option.

- **color-ramp-style-element**  
specifies the name of a color-ramp style element.
  
The style element should contain these style attributes:

  **STARTCOLOR**  
specifies the color for the smallest data value of the COLORRESPONSE= column.

  **NEUTRALCOLOR**  
specifies the color for the midpoint of the range of the COLORRESPONSE= column.
ENDCOLOR
specifies the color for the highest data value of the COLORRESPONSE= column.

(color-list)
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

See “color ” on page 1480

Default The ThreeColorRamp style element for filled bubbles and ThreeColorAltRamp for unfilled bubbles

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the numeric column or range attribute map variable to use to determine the bubble colors.

range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“RANGEATTRVAR Statement”

Restriction A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Interactions Prior to SAS 9.4M5, when the GROUP= option is specified with the COLORRESPONSE= option, the GROUP= option is ignored. Starting with SAS 9.4M5, the COLORRESPONSE= option controls the color of the group fill patterns when the fill patterns are displayed.

When fill is displayed, this option overrides suboption COLOR= in the FILLATTRS= option and varies the fill color according to the color gradient or the attribute map.

When only the outlines are displayed, this option overrides suboption COLOR= in the OUTLINEATTRS= option and varies the outline color according to the color gradient or the attribute map.
Notes
When both the fill and outline are displayed, the bubble fill color varies according to the color gradient or attribute map but the bubble outline color remains fixed on the color specified in option OUTLINEATTRS=.

The mapped color used for the bubbles is also used for the data labels.

Tip
To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

DATALABEL=column | expression
specifies a column for bubble labels. The label positions are adjusted to prevent them from overlapping.

Default
No labels are displayed.

Note
The BUBBLEPLOT statement does not use a data-label collision avoidance algorithm. Data labels in bubble plots might overlap.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the bubble labels.

Defaults
For non-grouped data, the GraphDataText style element.

For grouped data, the text color is derived from the GraphData1–GraphDataN style elements. The data label color changes to match the group color derived from the ContrastColor attribute of the style element that is in effect.

Interaction
The default attributes are overridden if the COLORRESPONSE= option is used.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

DATALABELPOSITION=TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the bubble labels relative to the bubble.

Default
TOPRIGHT

DATALABELSPLIT=TRUE | FALSE
specifies whether to split the data labels at the specified split characters. When set to TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters.

Default
FALSE. The data labels are not split.

Requirement
The DATALABEL= option must also be specified.

Interaction
The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

See
“boolean ” on page 1479 for other Boolean values that you can use.
**DATALABELSPLITCHAR=**"character-list"

specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

**Default**

A blank space

**Requirements**

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

datalabelsplitchar="abc"

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction**

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=**TRUE | FALSE

specifies whether the split characters are included in the data labels.

TRUE

drops the split characters from the data label.

FALSE

includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:

- the data label text for this label is Product*Group*A
- DATALABELSPLIT=TRUE
**DATALABELSPLITCHARDROP**=TRUE | FALSE

**DATALABELSPLITCHAR**="*"

When **DATALABELSPLITCHARDROP**=TRUE, the asterisks are removed from the label. When **DATALABELSPLITCHARDROP**=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**

TRUE. The split characters are dropped from the data label.

**Requirement**
The **DATALABEL=** option and the **DATALABELSPLIT**=TRUE option must also be specified.

**Interaction**
The **DATALABELSPLITCHAR**= option specifies the split characters.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**DATALABELSPLITJUSTIFY**=AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the data label blocks.

**AUTO**
justifies the labels based on the **DATALABELPOSITION=** option.

<table>
<thead>
<tr>
<th><strong>DATALABELPOSITION=</strong> Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

**CENTER**

**LEFT**

**RIGHT**

justifies the labels center, left, or right, as specified.

The following figure shows an example in which **DATALABELPOSITION=TOP**.

**Note:** The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.
In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

**Default**
AUTO

**Requirement**
The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**
enhances the visual appearance of the filled bubbles. The following figure shows bubbles with each of the skins applied.

<table>
<thead>
<tr>
<th>NONE</th>
<th>CRISP</th>
<th>GLOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MATTE</th>
<th>PRESSED</th>
<th>SHEEN</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
</tbody>
</table>

**Default**
The DATASKIN= option value that is specified in the BEGINGRAPH statement. If not specified, then the GraphSkins:DataSkin style element value is used.

**Restriction**
Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

**Requirement**
For this option to have any effect, DISPLAY= FILL must be in effect. Otherwise, this option is ignored.

**Interactions**
This option overrides the BEGINGRAPH statement DATASKIN= option.

The appearance of the data skin is based on the FILLATTRS= color.

This option is ignored if the RELATIVESCALE= option is set to FALSE.

When a data skin is applied, all bubble outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

**DATATRANSPARENCY=number**
specifies the degree of the transparency of the bubble fills and bubble outlines.
Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

Note This option does not affect the data labels.

Tip The FILLATTRS= option can be used to set transparency for just the filled bubble areas. You can combine this option with FILLATTRS= to set one transparency for the bubble outlines but a different transparency for the bubble fills. Example:

datatransparency=0.2 fillattrs=(transparency=0.6)

DISPLAY=(display-options) | STANDARD | ALL

specifies which bubble features to display.

(display-options)

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

OUTLINE
displays the bubble outline

FILL
displays the bubble fill color

FILLPATTERN
displays the bubble fill pattern

Note This feature applies to **SAS 9.4M5** and to later releases.

Tip By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the FILLPATTERNATTRS= option to specify a fill pattern for the bubbles.

STANDARD

specifies FILL and OUTLINE

ALL

specifies all features: FILL, FILLPATTERNATTRS, and OUTLINE

Default STANDARD

Tip Use the DATASKIN=, OUTLINEATTRS=, and FILLATTRS= options to control the appearance of the bubble outline and fill color. Starting with **SAS 9.4M5**, use the FILLPATTERNATTRS= option to control the appearance of the bubble fill pattern.

DRAWORDER=SIZE | DATA

specifies whether the bubbles are drawn according to bubble size or according to data order.

Note: This feature applies to **SAS 9.4M3** and to later releases.

SIZE
draws the bubbles according to bubble size, from the largest to the smallest
DATA
draws the bubbles according to data order

The following figure shows the effect of SIZE and DATA on four bubbles. The
bubble labels indicate the data order, and the bubble sizes increase linearly
starting with 1.

Default  SIZE

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the filled bubble areas.

Defaults  For non-grouped data, the GraphDataDefault:Color style reference.

For grouped data, the Color attribute of GraphData1–GraphDataN
style elements.

Interactions  For this option to have any effect, the fill must be enabled by the
ODS style or the DISPLAY= option.

When COLORRESPONSE= is in effect and the DISPLAY= option
enables FILL display, the FILLATTRS= suboption COLOR= is
ignored, and the bubble fill colors vary according to the gradient.

Tip  The DATATRANSPARENCY= option sets the transparency for the
bubble fills and the bubble outlines. You can combine this option
with DATATRANSPARENCY= to set one transparency for the
outlines but a different transparency for the fills. Example:
datatransparency=0.2 fillattrs=(transparency=0.6)

See  “General Syntax for Attribute Options” on page 1519 for the syntax
on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

Note:  This feature applies to SAS 9.4M5 and to later releases.

Defaults  For non-grouped data, the ContrastColor and FillPattern attributes
of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of
GraphData1–GraphDataN.

Interactions  For this option to take effect, the DISPLAY= option must include
FILLPATTERN among the display options.
Starting with SAS 9.4M5, when COLORRESPONSE= is in effect and the DISPLAY= option enables FILLPATTERN display, the FILLPATTERNATTRS= suboption COLOR= is ignored, and the fill-pattern colors vary according to the gradient.

**Tip**

In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in *SAS Graph Template Language: User’s Guide*.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Fill Pattern Options” on page 1521 for available *fill-pattern-options*.

GROUP= column | discrete-attr-var | expression
creates a separate bubble color for each unique grouping that is specified.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction**

A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

A distinct bubble is created for each group value by varying the visual attributes of the bubble display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature¹</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern ²</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern.</td>
</tr>
<tr>
<td>Display Feature</td>
<td>Style Attributes That Control Default Group Appearance</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Outline</td>
<td>if the outline is enabled with fill color, fill pattern, or both, ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style</td>
</tr>
<tr>
<td></td>
<td>if the outline is the only display feature enabled, ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element</td>
</tr>
</tbody>
</table>

1 Fill color and the outline are displayed by default.
2 Fill patterns are valid for bubble plots starting with SAS 9.4M5.

Interactions

If a discrete attribute map variable is specified, then the color mapping for the bubbles is defined by the associated DISCRETEATTRMAP statement. See "DISCRETEATTRMAP Statement" on page 1425.

The mapped color that is used for outlines is also used as the color of the data labels.

This option is ignored if the COLORRESPONSE= option is also used.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Notes

All bubbles have only one fill and one outline color as specified by the FILLATTRS= and OUTLINEATTRS options.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

See "DISCRETEATTRVAR Statement" on page 1435

**INCLUDEMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the group variable are included in the plot.

Default 

| TRUE |

Interaction

For this option to take effect, the GROUP= option must also be specified.

Tip

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is
applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See “boolean” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**
specifies indices for mapping bubble attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**Requirements**
The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Notes**
The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip**
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL=”string”**
specifies a label to be used in a discrete legend for this plot.

**Default**
The SIZE= column label. If a label is not defined, then the SIZE= column name is used.

**Restriction**
This option applies only to an associated DISCRETELEGEND statement.

**Interaction**
If the GROUP= option is specified, then this option is ignored.

**NAME=”string”**
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**
The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.
Interaction
The string is used as the default legend label if the LEGENDLABEL= option is not used.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the bubble outlines.

Defaults
For non-grouped data, the ContrastColor and LineStyle attributes of the GraphOutlines style element.

For grouped data and filled bubbles, the ContrastColor attribute of the GraphData1–GraphDataN style elements and a solid line style. If the COLORRESPONSE= option is specified, then the bubble outline attributes are derived from the GraphDataDefault style element.

For grouped data and unfilled bubbles, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements.

Interactions
For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

If the DATASKIN= option applies a data skin, then this option is ignored.

When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the bubble outline colors vary according to the gradient.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

RELATIVESCALE=TRUE | FALSE
specifies whether the SIZE= column values are interpreted as relative values. Relative means that the size values do not translate directly into bubble radii.
Rather, the bubble sizes are scaled to represent the value range of the SIZE=

For example, when RELATIVESCALE=TRUE, if only two bubbles are drawn with
sizes of 2 and 4, then they appear the same as only two bubbles with sizes of
4000 and 8000. By contrast, when RELATIVESCALE=FALSE, the size values
are interpreted in the same units as the axes.

If you set this option to FALSE, then it is recommended that you also place the
BUBBLEPLOT statement in a LAYOUT OVERLAYEQUATED container. This
ensures that the X axis and Y-axis units are the same. For more information, see
“Details” on page 431.

Default

<table>
<thead>
<tr>
<th>RELATIVESCALE=TRUE</th>
</tr>
</thead>
</table>

Interactions

When this option is set to TRUE, the BUBBLERADIUSMAX= and
BUBBLERADIUSMIN= options can be used to fix the drawing size
of the smallest bubble and largest bubble. If
RELATIVESCALE=FALSE, then the BUBBLERADIUSMAX= and
BUBBLERADIUSMIN= options are ignored.

If this option is set to FALSE, then the DATASKIN= option is
ignored.

If one or both axes are discrete, then RELATIVESCALE=FALSE is
ignored.

Tip

If you specify RELATIVESCALE=FALSE, then it is recommended
that you also place the BUBBLEPLOT statement in a LAYOUT
OVERLAYEQUATED container to ensure that the X axis and Y-axis
units are the same. If you place the BUBBLEPLOT statement in a
LAYOUT OVERLAY container instead, then the bubbles might be
drawn as ellipses because the X axis and Y-axis units are different.

See

“boolean ” on page 1479 for other Boolean values that you can use.

**RELATIVESCALETYPE=LINEAR | PROPORTIONAL**

specifies the type of scaling that is to be applied to the SIZE= column values.

**LINEAR**

increases the size of the bubbles in linear proportion to the range of the
SIZE= column values.

For example, if only two bubbles are drawn with sizes of 2 and 4, then they
appear the same as only two bubbles with sizes of 4000 and 8000.

**PROPORTIONAL**

increases the size of each bubble in direct proportion to its corresponding
SIZE= column value.

For example, if only two bubbles are drawn with sizes of 50 and 100, then the
bubble for SIZE=50 is drawn to half the size of the bubble for SIZE=100.

Default

<table>
<thead>
<tr>
<th>LINER</th>
</tr>
</thead>
</table>

Interactions

This option is ignored when RELATIVESCALE=FALSE.

When the SIZE= threshold value is greater than the
SIZE= column value, the proportional scale is adjusted.
so that the size of the bubble for that value is clamped to the BUBBLERADIUSMAX= value.

If all the values for the SIZE= column are negative, then RELATIVESCALETYPE=PROPORTIONAL is ignored, and the default value is used.

When RELATIVESCALETYPE=PROPORTIONAL is specified, the BUBBLERADIUSMIN= option specifies the minimum bubble size. In that case, when a SIZE= column value results in a bubble of a size that is less than the BUBBLERADIUSMIN= value, the bubble size for that value is changed to the BUBBLERADIUSMIN= value.

REVERSECOLORMODEL=TRUE | FALSE

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

Default FALSE

See COLORMODEL= “boolean ” on page 1479 for other Boolean values that you can use.

ROLENAMES=(role-name-list)

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

(role-name-list) a space-separated list of role-name = column pairs.

Example The following example assigns the column Obs to the user-defined role TIP:

ROLENAMES=(TIP1=OBS)

Default No user-defined roles

Requirement The role names that you choose must be unique and different from the predefined roles X, Y, SIZE, GROUP, DATALABEL, and COLORRESPONSE.

SIZETHRESHOLDMAX=numeric-value

specifies a SIZE= column value threshold at which bubble size is clamped to the BUBBLERADIUSMAX= option value. The size of the bubbles for all SIZE= column values that equal or exceed the specified threshold value is set to the BUBBLERADIUSMAX= value.

Default The maximum SIZE= column value is mapped to the BUBBLERADIUSMAX= option value.

TIP=(role-list) | NONE

specifies the information to display when the cursor is positioned over the bubbles. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the bubble plot can be specified along with roles that do.
(role-list)
an ordered, space-separated list of unique BUBBLEPLOT and user-defined roles.

BUBBLEPLOT roles include X, Y, SIZE, GROUP, DATALABEL, and COLORRESPONSE.

Tip User-defined roles are defined with the ROLENAMESPACE= option.

Example The following example displays data tips for the columns assigned to the roles X, Y, and SIZE, as well as the column Pop_2009. The POP_2009 column is not assigned to any pre-defined BUBBLEPLOT role, so it must first be assigned a role:

ROLENAMESPACE=(TIP1=POP_2009)
TIP=(TIP1 X Y SIZE)

NONE
suppresses data tips and URLs (if requested) from the plot.

Default The columns assigned to these roles are automatically included in the data tip information: X, Y, SIZE, GROUP, DATALABEL, and COLORRESPONSE.

Requirement To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example ROLENAMESPACE=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction Only the roles that appear in the TIP= option are used.

Requirement A column must be assigned to each of the specified roles. (See the ROLENAMESPACE= option.)

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.
**role-label-list**

a space-separated list of `rolename = "string"` pairs.

**Example**

```plaintext
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

**Default**

The column label or column name of the column assigned to the role.

**Restriction**

Only the roles that appear in the TIP= option are used.

**Requirement**

A column must be assigned to each of the specified roles. (See the `ROLENAME=` option.)

**URL=string-column**

specifies an HTML page to display when a bubble is selected.

**string-column**

specifies a column that contains a valid HTML page reference (HREF) for each bubble that is to have an active link.

**Example**

`http://www.sas.com`

**Requirement**

To generate a plot with selectable bubbles, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interactions**

This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tips**

The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding point is selected.

The URL value can be the same for any X and Y pairs. In that case, the same action is taken when the bubbles for those X and Y pairs are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default**

`X`
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=** | **Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Default**

Y

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**Details**

The BUBBLEPLOT statement displays one bubble for each row in the data, provided that row contains nonmissing values for X, Y, and SIZE. By default, the bubbles are displayed as filled, outlined circles. Regardless of the data order, the bubbles are always drawn from the largest size to the smallest size.

By default, the minimum and maximum values of the SIZE= column establish a range over which the bubble radii increase in linear proportion. The actual drawing size of the smallest and largest bubble is set automatically. You can adjust the smallest and largest bubble sizes with the **BUBBLERADIUSMIN=** and **BUBBLERADIUSMAX=** options. In these cases where the bubble sizes are proportional to each other, the default setting **RELATIVESCALE=TRUE** is appropriate.

If the SIZE= values are in the same units as the X and Y values, and both X and Y are numeric, then you can generate a plot where the bubble-radius units match the axis-scale units. To do so, specify the BUBBLEPLOT statement within a LAYOUT OVERLAYEQUATED block, and in the BUBBLEPLOT statement, set **RELATIVESCALE=FALSE**.

By default, for character columns, the X and Y axes are always discrete. For numeric columns, the X and Y axes are linear. You can change axis type for numeric axes with the layout options **XAXISOPTS=** and **YAXISOPTS=**.

**Note:** Within a LAYOUT OVERLAY, the unit-interval of the X and Y axes are not necessarily the same and the bubbles might be distorted into ellipses when **RELATIVESCALE=FALSE**. The OVERLAYEQUATED container ensures that the bubbles are displayed as circles, assuming that both the X= and Y= arguments specify numeric columns.

data influence;
  input x y radius category;
datalines;
  2 4 1 1
  5 5 2 1
  6 3 2 2
  12 7 3 2
;  
proc template;
  define statgraph equatedbubbles;
    begingraph;

```plaintext
```

```plaintext
```
Example: BUBBLEPLOT Statement

The following graph was generated by the "Example Program" on page 433.
Example Program

data bubbleintro;
input Engineer $ Salary number;
format Salary dollar7.0 number comma6.0;
datalines;
Electric 59000 89382
Civil 54000 73273
Software 56000 34833
Chemical 62000 25541
Mechanical 60000 19601
;
proc template;
define statgraph engineer;
begingraph;
entrytitle 'Median Salary for Entry Level Engineers';
entrytitle 'Bubbles Show Number of Engineers in Survey';
layout overlay;
bubbleplot x=engineer y=salary
   size=number / datalabel=number;
endlayout;
endgraph;
end;
proc sgrender data=bubbleintro template=engineer;
run;
CONTOURPLOTPARM Statement

Creates a contour plot representing a response variable evaluated over a grid of X and Y values.

Restriction:  Contour plots do not support data tips.

Tips:  By default, the CONTOURPLOTPARM statement assumes that the X-Y grid is complete and does not contain any missing or irregular values. If the X-Y grid is not complete, specify GRIDDED=FALSE in the CONTOURPLOTPARM statement so that the values needed to complete the grid are calculated. Otherwise, unexpected results might occur.

You can use a legend to display the contour level values. For CONTOURTYPE=LINE and CONTOURTYPE=LABLEDLINE, use a DISCRETELEGEND statement to add a legend. For all other contour types, use a CONTINUOUSLEGEND statement to add a legend.

For filled contour types, there might be small, visible gaps between the axes and the contour boundaries. To eliminate the gaps, specify the following axis options in the layout statement for the plots parent layout:

```
XAXISOPTS=(OFFSETMIN=0 OFFSETMAX=0 LINEAROPTS=(THRESHOLDMIN=0 THRESHOLDMAX=0))
YAXISOPTS=(OFFSETMIN=0 OFFSETMAX=0 LINEAROPTS=(THRESHOLDMIN=0 THRESHOLDMAX=0))
```

Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

```
CONTOURPLOTPARM X=numeric-column | expression
Y=numeric-column | expression
Z=numeric-column | expression </options>;
```

Summary of Optional Arguments

Appearance options

- COLORMODEL=style-element | (color-list)
  specifies a color ramp that is to be used to determine the colors of filled or gradient contours.

- CONTOURTYPE=LINE | FILL | GRADIENT | LINEFILL | LINEGRADIENT | LABLEDLINE | LABLEDLINEFILL | LABLEDLINEGRADIENT
  specifies how the contour is displayed.

- GRIDDED=TRUE | FALSE
  specifies whether the X and Y values are equally spaced in a rectangular grid.

- LEVELS=(contour-value-list)
  specifies a list of contour level values.

- LINEATTRS=style-element | style-element (line-options) | (line-options)
CONTOURPLOTPARM Statement 435

specifies the attributes of the contour lines.

\textbf{NHINT=integer}
specifies the suggested number of contour levels for the Z column.

\textbf{NLEVELS=integer}
specifies the actual number of contour levels for the Z column.

\textbf{REVERSECOLORMODEL=TRUE \mid FALSE}
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

\textbf{Axes options}

\textbf{PRIMARY=TRUE \mid FALSE}
specifies that the data columns for this plot and the plot type be used for determining default axis features.

\textbf{XAXIS=X \mid X2}
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

\textbf{YAXIS=Y \mid Y2}
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

\textbf{Label options}

\textbf{LEGENDLABEL="string"}
specifies a label to be used in a discrete legend for this plot.

\textbf{LINELABELATTRS=style-element \mid style-element (text-options) \mid (text-options)}
specifies the color and font attributes of the contour line labels.

\textbf{LINELABELBASELINE=HORIZONTAL \mid TANGENT}
specifies the text alignment of the contour line labels.

\textbf{LINELABELFORMAT=format}
specifies the format to use for the contour line labels.

\textbf{LINELABELPOSITION=MIDDLE \mid BEGIN \mid END}
specifies the position for the contour line labels.

\textbf{Plot reference options}

\textbf{NAME="string"}
assigns a name to this plot statement for reference in other template statements.

\textbf{Required Arguments}

\textbf{X=numeric-column \mid expression}
specifies the X coordinates for the grid.

\textbf{Y=numeric-column \mid expression}
specifies the Y coordinates for the grid.

\textbf{Z=numeric-column \mid expression}
specifies the contour response values.

\textbf{Optional Arguments}

\textbf{COLORMODEL=style-element \mid \{color-list\}}
specifies a color ramp that is to be used to determine the colors of filled or gradient contours.
**style-element**

specifies the name of a style element. The style element can contain these style attributes:

- **STARTCOLOR** specifies a color for the smallest data value of the Z= column.
- **NEUTRALCOLOR** specifies a color for the midpoint of the range of the Z= column.
- **ENDCOLOR** specifies a color for the highest data value of the Z= column.

**color-list**

specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData2:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement**

The list of colors must be enclosed in parentheses.

**See**

“color ” on page 1480

**Default**

The ThreeColorRamp style element.

**Interactions**

The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

The NEUTRALCOLOR attribute is not used for two-color ramps.

**Note**

For a contour plot of type FILL, LINEFILL, or LABELEDLINEFILL, the color that is chosen for the first segment is the color for the N+1 inflection point and not the starting color in the color model.

**CONTOURTYPE=**

LINE | FILL | GRADIENT | LINEFILL | LINEGRADIENT | LABELEDLINE | LABELEDLINEFILL | LABELEDLINEGRADIENT

specifies how the contour is displayed.

- **LINE**
  
displays contour levels as unlabeled lines.

- **FILL**
  
displays the area between the contour levels as filled.

  Each contour interval is filled with one color.

- **GRADIENT**
  
displays a smooth gradient of color to represent contour levels.

- **LINEFILL**
  
combines the LINE and FILL types.

  Each contour interval is filled with one color. Displays contour levels as unlabeled lines.

- **LINEGRADIENT**
  
combines the LINE and GRADIENT types.

  Displays contour levels as unlabeled lines.
LABELEDLINE
adds labels to the LINE type, displaying contour levels as labeled lines.

LABELEDLINEFILL
adds labels to the LINEFILL type.
Each contour interval is filled with one color. Displays contour levels as lines with labels showing contour level values.

LABELEDLINEGRADIENT
adds labels to the LINEGRADIENT type.
Displays contour levels as lines with labels showing contour level values.

The following figure shows the effect of each of the values.

<table>
<thead>
<tr>
<th>LINE</th>
<th>FILL</th>
<th>GRADIENT</th>
<th>LINEFILL</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Line" /></td>
<td><img src="image2" alt="Fill" /></td>
<td><img src="image3" alt="Gradient" /></td>
<td><img src="image4" alt="LineFill" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LINEGRADIENT</th>
<th>LABELEDLINE</th>
<th>LABELEDLINEFILL</th>
<th>LABELEDLINEGRADIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5" alt="LineGradient" /></td>
<td><img src="image6" alt="LabeledLine" /></td>
<td><img src="image7" alt="LabeledLineFill" /></td>
<td><img src="image8" alt="LabeledLineGradient" /></td>
</tr>
</tbody>
</table>

**Default**
The GraphContour:DisplayOpts style reference.

**Interactions**
The fill colors of the types that enable FILL or GRADIENT are controlled by the `COLORMODEL=` option.

The line properties of the types that enable LINE or LABELEDLINE are controlled by the `LINEATTRS=` option.

The label properties of the types that enable LABELEDLINE are controlled by the `LINELABELATTRS=` and `LINELABELBASELINE=` options.

If a DISCRETELEGEND statement is associated with the contour, then the legend is NOT displayed if `CONTOURTYPE=` is set to FILL or GRADIENT.

If a CONTINUOUSLEGEND statement is associated with the contour, then the legend is NOT displayed if `CONTOURTYPE=` is set to LINE or LABELEDLINE.

**Note**
For a contour plot of type FILL, LINEFILL, or LABELEDLINEFILL, the color that is chosen for the first segment is the color for the N+1 inflection point and not the starting color in the color model.

**GRIDDED=TRUE | FALSE**
specifies whether the X and Y values are equally spaced in a rectangular grid. If set to FALSE, then additional calculations are performed in order to complete the grid.
For information about the algorithm used to calculate the grid, see "Mesoscale Objective Map Analysis Using Weighted Time-Series Observations."1

**Default** TRUE

**Tip** By default, the CONTOURPLOTPARM statement assumes that the X-Y grid is complete and does not contain any missing or irregular values. If the X-Y grid is not complete, then specify GRIDDED=FALSE so that the plot calculates the values needed to complete the grid. Otherwise, unexpected results might occur.

**See** “boolean” on page 1479 for other Boolean values that you can use.

**LEGENDLABEL=“string”**

specifies a label to be used in a discrete legend for this plot.

**Default** The Z-column label. If a label is not defined, then the Z-column name is used.

**Restrictions** This option applies only to an associated DISCRETELEGEND statement.

This option applies to contour types LINE and LabeledLINE only.

**LEVELS=(contour-value-list)**

specifies a list of contour level values.

**(contour-value-list)**

a space-separated list of numeric values, enclosed in parentheses.

**Default** The number of levels and the level values are determined internally using the NHINT= or NLEVELS= option value.

**Interaction** This option overrides the NHINT= and NLEVELS= options.

**Notes** Values that are outside of the data range are ignored.

When the LEVELS= option is not specified, the data range might be extended in order to accommodate the internally computed levels.

**Example** The following example specifies six levels for a contour plot:
levels=(0.0001 0.0004 0.0007 0.0010 0.0013 0.0016)

**LINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the contour lines.

**Default** The GraphDataDefault style element.

**Interaction** This option is honored only if the CONTOURTYPE= displays lines.

**See** “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.
LINELABELATRIVS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the contour line labels.

Default            The GraphValueText style element.
Interaction         This option is honored only if the CONTOURTYPE= displays labels.
See                 "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.
                     “Text Options” on page 1525 for available text-options.

LINELABELBASELINE=HORIZONTAL | TANGENT
specifies the text alignment of the contour line labels. Each contour line has a precomputed label point.

HORIZONTAL
    specifies that each label is parallel to the X axis.
    The label intersects its contour line and is centered at the label point.

TANGENT
    specifies that each label is drawn tangent to the contour line at the label point.
    This reduces intersection with the contour line and causes labels to be tilted at various angles in relation to the X axis.

Default            HORIZONTAL
Interaction         This option is honored only if the CONTOURTYPE= displays labels.

LINELABELFORMAT=format
specifies the format to use for the contour line labels.

Default            The format associated with the Z column or BEST6. if no format is assigned.
Interaction         This option is honored only if the CONTOURTYPE= displays labels.

LINELABELPOSITION=MIDDLE | BEGIN | END
specifies the position for the contour line labels.

Default            MIDDLE
Interaction         This option is honored only if the CONTOURTYPE= option specifies labels.

NAME="string"
assigns a name to this plot statement for reference in other template statements.
The specified name is used primarily in legend statements to indicate the colors associated with the Z values.

Restrictions       The string is case sensitive, cannot contain spaces, and must define a unique name within the template.
                     This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.
Interaction  The string is used as the default legend label if the LEGENDLABEL= option is not used.

**NHINT=integer**

specifies the suggested number of contour levels for the Z column.

- Default: 7
- Interaction: This option is ignored if the LEVELS= or NLEVELS= option is specified.
- Note: The actual number of levels is adjusted to provide an appropriate number of levels for the data.

**NLEVELS=integer**

specifies the actual number of contour levels for the Z column.

- Default: The number of levels is determined internally, using the NHINT= value.
- Interaction: This option overrides the NHINT= option.
- This option is ignored if CONTOURTYPE= GRADIENT.
- This option is ignored if the LEVELS= option is specified.

**PRIMARY=TRUE | FALSE**

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

- Default: FALSE
- Restriction: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.
- Note: In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.
- See: “When Plots Share Data and a Common Axis” on page 979

**REVERSECOLORMODEL=TRUE | FALSE**

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

- Default: FALSE
- See: COLORMODEL= "boolean " on page 1479 for other Boolean values that you can use.
**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

### Details

A CONTOURPLOTPARM statement uses the `CONTOURTYPE=` option to specify the type of contour plot to generate. Contour types that display fills or gradients but no contour lines can use only a CONTINUOUSLEGEND statement to represent the contour level values in a legend. Contour types that display lines can use either a CONTINUOUSLEGEND or DISCRETELEGEND statement to identify contour level values.

By default, the CONTOURPLOTPARM statement assumes that the X-Y grid is complete. If the grid is not complete, then set the `GRIDDED=` option to FALSE so that the plot calculates the values needed to complete into a grid with a bounding rectangle.

For filled or gradient contour types, small gaps might be visible between the axes and the bounding box of the contour data. To eliminate these gaps, set the axis options of the LAYOUT OVERLAY statement as follows:

```bash
XAXISOPTS=(OFFSETMIN=0 OFFSETMAX=0 LINEAROPTS=(THRESHOLDMIN=0 THRESHOLDMAX=0))
YAXISOPTS=(OFFSETMIN=0 OFFSETMAX=0 LINEAROPTS=(THRESHOLDMIN=0 THRESHOLDMAX=0))
```

Contour plots do not support the data tips that are enabled by the IMAGEMAP= option in the ODS GRAPHICS statement.

### Example: CONTOURPLOTPARM Statement

The following graph was generated by the “Example Program” on page 442:
Example Program

```sas
proc template;
define statgraph contourplotparm;
begingraph;
  entrytitle "Contour Plot of Height and Weight";
  layout overlay /
    xaxisopts=(offsetmin=0 offsetmax=0
               linearopts=(thresholdmin=0 thresholdmax=0))
    yaxisopts=(offsetmin=0 offsetmax=0
               linearopts=(viewmax=250
                           thresholdmin=0 thresholdmax=0));
  contourplotparm x=height y=weight z=density /
    contourtype=fill nhint=12
    name="Contour" colormodel=twocolorramp;
  continuouslegend "Contour" / title="Density";
endlayout;
endgraph;
end;
run;
proc sgrender data=sashelp.gridded template=contourplotparm;
run;
```

DENDROGRAM Statement

Creates a tree diagram that is typically used to display the results of a hierarchical clustering analysis.

Restriction: The DENDROGRAM statement does not support CAS in-memory tables. In SAS 9.4M6 and earlier releases, unexpected results might occur when data stored in a CAS in-
memory table is used. Starting with SAS Viya 3.5 and SAS Studio 5.2, the dendrogram is dropped when data stored in a CAS in-memory table is used.

Syntax

**DENDROGRAM** NODEID=\( \text{column} \mid \text{expression} \)
PARENTID=\( \text{column} \mid \text{expression} \)
CLUSTERHEIGHT=\( \text{numeric-column} \mid \text{expression} \langle \text{options} \rangle \);

Summary of Optional Arguments

**Appearance options**

- **CLUSTERS=**\( \text{numeric-column} \mid \text{expression} \)
  - Specifies a numeric column containing the resultant number of clusters at each node.
- **CUT=**\( \text{TRUE} \mid \text{FALSE} \)
  - Specifies whether the tree is to be cut.
- **CUTOPTS=**\( \text{(pruning-options)} \)
  - Specifies pruning options for cutting the dendrogram.
- **DATATRANSPARENCY=**\( \text{number} \)
  - Specifies the degree of the transparency of the dendrogram lines.
- **LINEATTRS=**\( \text{style-element} \mid \text{style-element} \langle \text{line-options} \rangle \mid \langle \text{line-options} \rangle \)
  - Specifies the attributes of the dendrogram lines.
- **ORIENT=**\( \text{VERTICAL} \mid \text{HORIZONTAL} \)
  - Specifies the orientation of the dendrogram leaf axis.
- **TREETYPE=**\( \text{RECTANGULAR} \mid \text{TRIANGULAR} \)
  - Specifies the type of tree structure to draw.

**Axes options**

- **PRIMARY=**\( \text{TRUE} \mid \text{FALSE} \)
  - Specifies that the data columns for this plot and the plot type be used for determining default axis features.
- **XAXIS=**\( \text{X} \mid \text{X2} \)
  - Specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
- **YAXIS=**\( \text{Y} \mid \text{Y2} \)
  - Specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Data tip options**

- **TIP=**\( \text{(role-list)} \mid \text{NONE} \)
  - Specifies the information to display when the cursor is positioned over a dendrogram line.
- **TIPFORMAT=**\( \text{(role-format-list)} \)
  - Specifies display formats for tip columns.
- **TIPLABEL=**\( \text{(role-label-list)} \)
  - Specifies display labels for tip columns.

**Label options**

- **LEGENDLABEL=**\( \text{"string"} \)
specifies a label to be used in a discrete legend for this plot.

**Plot reference options**

NAME="string"

assigns a name to this plot statement for reference in other template statements.

**Required Arguments**

**NODEID=column | expression**

specifies a column for the ID values of the nodes. Each node ID value must be unique. If duplicate NODEID values are found, then the dendrogram is not rendered. The column can be numeric or character, but it must be of the same type and have the same formatted length as the ParentID column.

The maximum number of nodes that are supported by the dendrogram is determined by the DISCRETEMAX= option in the ODS GRAPHICS statement. The default value is DISCRETEMAX=1000. If the graph data contains more than 1000 discrete values, then the dendrogram is not drawn and a warning is written to the SAS log. In that case, you can use the DISCRETEMAX= option to increase the maximum number of discrete values that are allowed.

**Note** If the data is a SAS data set, the nodes are displayed in data order. If the data is a CAS in-memory table, the nodes are displayed in ascending order of the NODEID values by default.

**PARENTID=column | expression**

specifies a column for the parent ID values of the nodes. The column can be numeric or character, but it must be of the same type and have the same formatted length as the NodeID column.

**CLUSTERHEIGHT=numeric-column | expression**

specifies the column for the height values for each node.

**Optional Arguments**

**CLUSTERS=numeric-column | expression**

specifies a numeric column containing the resultant number of clusters at each node.

**Interaction** For this option to take effect, the pruning options in the CUTOPTS= option must set TYPE=NCLUSTERS and specify a number for the NCLUSTERS= setting.

**CUT=TRUE | FALSE**

specifies whether the tree is to be cut.

**Default** FALSE

**Tip** To set the properties of the CUT, use the CUTOPTS= option.

**See** “boolean” on page 1479 for other Boolean values that you can use.

**CUTOPTS=(pruning-options)**

specifies pruning options for cutting the dendrogram. The following pruning-options must be specified as a space-separated list of option = value pairs enclosed in parentheses.
CUTHEIGHT=number
specifies the height at which the tree is to be pruned.

Default The tree is not pruned.

Requirement You must include pruning option TYPE=CUTHEIGHT with this option.

Interaction This option is ignored when CUT=FALSE or when pruning option TYPE=CUTHEIGHT is not explicitly specified.

Example CUTOPTS=(TYPE=CUTHEIGHT CUTHEIGHT=0.75)

NCLUSTERS=number
specifies the number of clusters to use for pruning the tree.

Default The tree is not pruned.

Interaction For this setting to take effect, pruning-option TYPE=NCLUSTERS must also be set. In addition, the CLUSTERS= option must be used, and the CUT= option must be set to TRUE.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the cut lines.

Default The GraphDataDefault style element.

See "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Line Options" on page 1522 for available line-options.

TYPE=CUTHEIGHT | NCLUSTERS
specifies which rule to use to prune the tree.

Default CUTHEIGHT

DATATRANSPARENCY=number
specifies the degree of the transparency of the dendrogram lines.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Default The CLUSTERHEIGHT= column label. If a label is not defined, then the CLUSTERHEIGHT= column name is used.

Restriction This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the dendrogram lines.

Default The GraphDataDefault style element.
NAME="string"

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction  The string is used as the default legend label if the LEGENDLABEL= option is not used.

ORIENT=VERTICAL | HORIZONTAL

specifies the orientation of the dendrogram leaf axis.

Default  VERTICAL

PRIMARY=TRUE | FALSE

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default  FALSE

Restriction  This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note  In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See  “When Plots Share Data and a Common Axis” on page 979

"boolean " on page 1479 for other Boolean values that you can use.

TIP=(role-list) | NONE

specifies the information to display when the cursor is positioned over a dendrogram line. If this option is used, then it replaces all of the information that is displayed by default.

(role-list)

an ordered, space-separated list of unique DENDROGRAM roles, which include NODEID, PARENTID, and CLUSTERHEIGHT.

Example  The following example displays data tips for the columns assigned to the roles NODEID and PARENTID.

TIP=(NODEID PARENTID)
NONE
suppresses data tips from the plot.

Default
The columns assigned to the following roles are automatically included in the data tip information: NODEID, PARENTID, and CLUSTERHEIGHT.

Requirement
To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip
The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example
TIPFORMAT=(CLUSTERHEIGHT=4.1)

Default
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles.

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename = "string" pairs.

Example
TIPLABEL=(CLUSTERHEIGHT="Height")

Default
The column label or column name of the column assigned to the role.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles.

TREETYPE=RECTANGULAR | TRIANGULAR
specifies the type of tree structure to draw.

Default
RECTANGULAR
**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default** X

**Interaction** The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

---

**YAXIS=Y | Y2**

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Default** Y

**Interaction** The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

---

**Details**

A dendrogram is a tree diagram that is typically used to show the cluster arrangements in hierarchical data. The DENDROGRAM statement supports clusters with only a single root. If multiple roots are found in the data, then a warning is written to the SAS log and the dendrogram is not drawn. The maximum number of nodes that is supported by the DENDROGRAM statement is determined by the DISCRETEMAX= option in the ODS GRAPHICS statement, which is 1000 by default.

In the Graph Template Language, a DENDROGRAM plot typically appears by itself in a LAYOUT OVERLAY container. You can overlay REFERENCELINE or BANDPLOT statements on a DENDROGRAM, but overlaying other plot types might produce unexpected results.

Using the DENDROGRAM statement in layouts where the axis ranges are merged across cells might produce unexpected results.

**Example: DENDROGRAM Statement**

The following graph was generated by the “Example Program” on page 449:
Example Program

data clustree;
  input id $ parent $7-12 height nClus;
  label id="Cluster ID" parent="Parent ID";
  datalines;
    clus1       3 1
    clus2 clus1 0.2 7
    clus3 clus1 1.75 2
    clus4 clus3 0.7 4
    clus5 clus3 0.8 3
    clus6 clus4 0.4 5
    clus7 clus6 0.1 9
    clus8 clus5 0.25 6
    clus9 clus8 0.15 8
    1  clus9 0 10
    2  clus6 0 10
    3  clus2 0 10
    4  clus7 0 10
    5  clus7 0 10
    6  clus2 0 10
    7  clus4 0 10
    8  clus5 0 10
    9  clus8 0 10
    10 clus9 0 10
run;

proc template;
  define statgraph dendrogram;
  begingraph;
DENSITYPLOT Statement

Creates a univariate probability density curve computed from input data.

Requirement: If you include a histogram and a density plot in your graph and both plots specify the same data column, you must either specify SCALE=DENSITY in the HISTOGRAM statement or place the HISTOGRAM statement before the DENSITYPLOT statement in your template code. Otherwise, the density plot values might not scale properly to the histogram bins on the dependent axis. This requirement applies even when the histogram and density plots are in different cells in a multicell graph.

Tips: If the data density is not known, then use the KERNEL distribution option in the DENSITYPLOT statement.

Starting with SAS 9.4M3, subpixel rendering is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

DENSITYPLOT numeric-column | expression </distribution-option> <options>;

Summary of Optional Arguments

Appearance options

- **DATATRANSPARENCY=number**
  specifies the degree of the transparency of the density curve and curve label.

- **LINEATTRS=style-element | style-element (line-options) | (line-options)**
  specifies the attributes of the density curve.

- **ORIENT=VERTICAL | HORIZONTAL**
  specifies the orientation of the Y axis.

Axes options

- **PRIMARY=TRUE | FALSE**
  specifies that the data columns for this plot and the plot type be used for determining default axis features.

- **XAXIS=X | X2**
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options
TIPFORMAT=(role-format-list)
specifies display formats for tip columns.
TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options
CURVELABEL="string"
specifies a label for the density curve.
CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the density curve label.
CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the density curve label relative to the plot area.
CURVELABELPOSITION=AUTO | MAX | MIN | START | END
specifies the position of the density curve label relative to the curve line.
CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the curve label at the specified split characters.
CURVELABELSPLITCHAR=character-list
specifies one or more characters on which the curve label can be split if needed.
CURVELABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the curve label text.
CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the curve label block.
LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options
GROUP=column | discrete-attr-var | expression
creates a separate density curve for each unique group value in the specified column.
INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Statistics options
FREQ=numeric-column | expression
specifies a numeric column that provides frequencies for each observation that is read.
WEIGHT=numeric-column | expression
specifies a column that contains a density-curve calculation a priori weight for each observation of the input data object.

Required Arguments

**numeric-column**
specifies a numeric column of data values that are used to calculate the parameters for the probability distribution.

**expression**
specifies an expression that calculates values when those values are not stored in the data.

Optional Arguments

**CURVELABEL=**"string"
specifies a label for the density curve.

- **Default**: No curve label is displayed.
- **Restriction**: This option is not valid when the **GROUP=** option is specified.
- **Tip**: The font and color attributes for the label are specified by the **CURVELABELATTRS=** option.

**CURVELABELATTRS=**style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the density curve label.

- **Default**: The GraphValueText style element.
- **Interaction**: For this option to take effect, the **CURVELABEL=curvelabel** option must also be used.
- **See**: “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

**CURVELABELLOCATION=**INSIDE | OUTSIDE
specifies the location of the density curve label relative to the plot area.

- **INSIDE**: locates the labels inside the plot area
- **OUTSIDE**: locates the labels outside the plot area

- **Default**: INSIDE
- **Restriction**: OUTSIDE cannot be used when the DENSITYPLOT is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes can be external to the grid.
- **Interaction**: This option is used in conjunction with the **CURVELABELPOSITION=** option to determine where the curve label appears. For more information, see “Location and Position of Curve Labels” on page 200.
CURVELABELPOSITION=AUTO | MAX | MIN | START | END
specifies the position of the density curve label relative to the curve line.

AUTO
positions the density label automatically near the end of the density curve along unused axes whenever possible (typically Y2 or X2) to avoid collision with tick values.

Restriction  This option is used only when CURVELABELLOCATION=OUTSIDE.

MAX
forces the density label to appear near maximum density X-values (typically, to the right).

MIN
forces the density label to appear near minimum density X-values (typically, to the left).

START
forces the density label to appear near the beginning of the curve.

Restriction  This option is used only when CURVELABELLOCATION=INSIDE.

Tip  This option is particularly useful when the curve line has a spiral shape.

END
forces the density label to appear near the end of the curve.

Restriction  This option is used only when CURVELABELLOCATION=INSIDE.

Tip  This option is particularly useful when the curve line has a spiral shape.

Defaults  AUTO when CURVELABELLOCATION=OUTSIDE.

END when CURVELABELLOCATION=INSIDE.

Restriction  The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

Interactions  For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELLOCATION= option to determine where the density label appears. For more information, see “Location and Position of Curve Labels” on page 200.

Note  When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the curve label might fall outside of the graph area. In that case, the curve label might not be displayed or might be positioned incorrectly.
**Tip**
When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.

**CURVELABELSPLIT=TRUE | FALSE**
specifies whether to split the curve label at the specified split characters. When a curve label is split, the label is split on each occurrence of the specified split characters.

- **Default**: FALSE. The curve label is not split.
- **Requirement**: The CURVELABEL= option must also be specified.
- **Interactions**: The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.
- **See**: “boolean” on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITCHAR="character-list"**
specifies one or more characters on which the curve label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the curve label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the curve label is split unconditionally at each occurrence of any of the specified split characters. If the curve label does not contain any of the specified characters, then the label is not split.

- **"character-list"**: one or more characters with no delimiter between each character and enclosed in quotation marks.
- **Default**: A blank space
- **Requirements**: The list of characters must be enclosed in quotation marks.
- **Interactions**: This option has no effect if CURVELABELPOSITION=AUTO.
- **Notes**: When multiple characters are specified, the order of the characters in the list is not significant.
Tip Use the CURVELABELSPLITJUSTIFY= option to specify the justification of the strings in the curve label block.

**CURVELABELSPLITCHARDROP=TRUE | FALSE**
specifies whether the split characters are included in the curve label text.

**TRUE**
drops the split characters from the curve label text.

**FALSE**
includes the split characters in the curve label text.

When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a curve label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

**Note:** The horizontal line to the left of the label represents the maximum end of the curve for reference.

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default** TRUE. The split characters are dropped from the curve label.

**Requirement** The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction** The CURVELABELSPLITCHAR= option specifies the split characters.

**See** “boolean” on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**
specifies the justification of the strings that are inside the curve label block.
AUTO
  justifies the labels based on the CURVELABELPOSITION= option.

<table>
<thead>
<tr>
<th>CURVELABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

CENTER
  LEFT
  RIGHT
  justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the curve for reference.

\[
\begin{array}{c|c|c|c}
\hline
\text{AUTO} & \text{CENTER} & \text{LEFT} & \text{RIGHT} \\
\hline
\text{Product Group A} & \text{Product Group A} & \text{Product Group A} & \text{Product Group A} \\
\hline
\end{array}
\]

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

Default  AUTO

Requirement  The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction  This option has no effect if CURVELABELPOSITION=AUTO.

DATATRANSPARENCY=number
  specifies the degree of the transparency of the density curve and curve label.

Default  0

Range  0–1, where 0 is opaque and 1 is entirely transparent

FREQ=numeric-column | expression
  specifies a numeric column that provides frequencies for each observation that is read.

Default  All observations have a frequency count of 1.

Restriction  If the value of the numeric-column is missing or is less than 1, then the observation is not used in the analysis. If the value is not an integer, then only the integer portion is used.
Note If \( n \) is the value of the numeric column for a given observation, then that observation is used \( n \) times for the purposes of any statistical computation.

**GROUP=** column | discrete-atr-va | expression
creates a separate density curve for each unique group value in the specified column.

Note: This feature applies to SAS 9.4M2 and to later releases.

discrete-atr-va
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default Each distinct group value is represented in the plot by a different line color and pattern. The line color is determined by the ContrastColor attribute of the GraphData1–GraphDataN and GraphMissing style elements. The line pattern is determined by the LineStyle attribute of the GraphData1–GraphDataN and GraphMissing style elements.

Notes The group values are mapped in the order in which they appear in the data.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

Tip You can individually override the representations that are used to identify the groups. For example, in some ODS styles, each distinct group value is represented by a different line color and pattern. In that case, you can use the PATTERN= setting on the LINEATTRS= option to assign the same line pattern to all of the curves.

See “DISCRETEATTRVAR Statement” on page 1435
“Example 3: Grouped Density Plot” on page 464

**INCLUDEMISSINGGROUP=** TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Note: This feature applies to SAS 9.4M2 and to later releases.

Default TRUE

Interaction For this option to take effect, the GROUP= option must also be specified.
Tip  The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**LEGENDLABEL=**"*string*"

specifies a label to be used in a discrete legend for this plot.

**Default**  The *string* specified on the **NAME=** option.

**Restriction**  This option applies only to an associated **DISCRETELEGEND** statement.

**LINEATTRS=** *style-element | style-element (line-options) | (line-options)*

specifies the attributes of the density curve.

**Defaults**  For non-grouped data, the GraphFit style element.

For grouped data, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the GraphFit:LineThickness style reference.

**See**  “General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

"Line Options" on page 1522 for available *line-options*.

**NAME=**"*string*"

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**  The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**  The *string* is used as the default legend label if the **LEGENDLABEL=** option is not used.

**ORIENT=** VERTICAL | HORIZONTAL

specifies the orientation of the Y axis.

**Default**  VERTICAL

**PRIMARY=** TRUE | FALSE

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**  FALSE
Restriction  This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note  In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See  “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example  TIPFORMAT=(Y=6.2)

Default  The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Requirement  To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Note  The columns assigned to the X and Y roles are automatically included in the data tip information.

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example  TIPLABEL=(Y="Curve")

Default  The column label or column name of the column assigned to the role.

Note  The columns assigned to the X and Y roles are automatically included in the data tip information.

WEIGHT=numeric-column | expression
specifies a column that contains a density-curve calculation a priori weight for each observation of the input data object.

Requirement  The value must be nonnegative.

Note  The values of the weight variable must be greater than zero. If the weight value for an observation is zero, negative, or missing, that observation is deleted from the analysis.
**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default** X

**Interaction** The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Default** Y

**Interaction** The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

**Distribution Options**

**KERNEL (<C=positive-number> <WEIGHTFUNCTION=normal | quadratic | triangular> <MAXPOINTS=positive-integer>)**
specifies a nonparametric kernel density estimate. The general form of the kernel density estimator is as follows.

\[
\hat{f}_h(x) = \frac{100h}{n\lambda} \sum_{i=1}^{n} K_0 \left( \frac{x - x_i}{\lambda} \right)
\]

In the equation, \( K_0(\cdot) \) is the weight function, \( \lambda \) is the bandwidth, \( n \) is the sample size, and \( x_i \) is the \( i \)th observation. You can use the \( C= \) suboption to specify the bandwidth and the \( WEIGHTFUNCTION= \) suboption to specify the weight function \( K_0(\cdot) \).

For more information, see the discussion of Kernel Density Estimates for the UNIVARIATE procedure in the documentation for Base SAS.

**TIP** Use the KERNEL distribution option when the data density is not known.

**C=positive-number**
specifies a positive number (0 < number <= 100) that represents the standardized bandwidth.

The value of \( \lambda \), referred to as the bandwidth parameter, determines the degree of smoothness in the estimated density function. You specify \( \lambda \) indirectly by specifying a standardized bandwidth \( c \) with the \( C= \) kernel-option. If \( Q \) is the interquartile range and \( n \) is the sample size, then \( c \) is related to \( \lambda \) by the following equation:

\[
\lambda = cQn - \frac{1}{5}
\]
Calculated from the data as the bandwidth that minimizes the approximate mean integrated square error (MISE).

Range 0 to 100 (inclusive)

**WEIGHTFUNCTION=**NORMAL | QUADRATIC | TRIANGULAR
 specifies one of the weight functions NORMAL, QUADRATIC, or TRIANGULAR.

The following table shows the formulas for the weight functions.

<table>
<thead>
<tr>
<th>Weight Function</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>( K_0(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right) ) for (-\infty &lt; t &lt; \infty)</td>
</tr>
<tr>
<td>QUADRATIC</td>
<td>( K_0(t) = \frac{3}{4}(1 - t^2) ) for (</td>
</tr>
<tr>
<td>TRIANGULAR</td>
<td>( K_0(t) = 1 -</td>
</tr>
</tbody>
</table>

Default NORMAL

**Note** In prior SAS releases, the weight function was specified with the WEIGHT= option. In SAS 9.4 and later releases, the WEIGHT= option is not valid as a distribution option. You must use the WEIGHTFUNCTION= option instead.

**MAXPOINTS=**positive-integer
 specifies the maximum number of points generated for the curve.

Default 512

Requirements
Prior to SAS 9.4M5, positive-integer must be greater than or equal to 2.
Starting with SAS 9.4M5, positive-integer must be greater than or equal to 16.

**NORMAL** (<**MU=**number> <**SIGMA=**number> <**MAXPOINTS=**number>)
specifies a normal density estimate, with mean and standard deviation. The fitted density function equation is as follows:

\[ p(x) = \frac{100\%}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x - \mu}{\sigma}\right)^2\right) \] for \(-\infty < x < \infty\)

In the equation, \(\mu\) is the mean and \(\sigma\) is the standard deviation (\(\sigma > 0\)). You can specify \(\mu\) with the MU= suboption and \(\sigma\) with the SIGMA= suboption. By default, ODS estimates \(\mu\) with the sample mean and \(\sigma\) with the sample standard deviation.

For more information, see the discussion of Kernel Density Estimates for the UNIVARIATE procedure in the documentation for Base SAS.

**MU=**number
 specifies the mean.
Default The value is calculated from the data.

SIGMA=number specifies the standard deviation.
Default The value is calculated from the data.

MAXPOINTS=positive-integer specifies the maximum number of points generated for the curve.
Default 200
Requirements Prior to SAS 9.4M5, positive-integer must be greater than or equal to 2.
Starting with SAS 9.4M5, positive-integer must be greater than or equal to 16.

Details
A typical DENSITYPLOT statement specifies either the NORMAL or the KERNEL distribution option. If no distribution option is specified, then the NORMAL() option is used. The following syntax explicitly shows the default case:

DENSITYPLOT numeric-column / NORMAL()

To specify a kernel distribution, use the following plot syntax:

DENSITYPLOT numeric-column / KERNEL()

If more than one distribution option is specified, then the last distribution option specified is used.

Examples:

Example 1: Overlaid Density Plots
The following graph was generated by the “Example Program” on page 463:
Example Program

When used as a stand-alone plot or overlaid with other density plots, the dependent axis shows the computed density values.

```sas
proc template;
    define statgraph densityplot1;
    begingraph;
        entrytitle "Fitted Density Curves";
        entrytitle "of Patient Weight";
        entryfootnote halign=left "Framingham Heart Study";
        layout overlay;
            densityplot weight / normal()
            lineattrs=graphfit name="n" legendlabel="Normal";
            densityplot weight / kernel()
            lineattrs=graphfit2 name="k" legendlabel="Kernel";
            discretelegend "n" "k";
        endlayout;
    endgraph;
end;
run;
proc sgrender data=sashelp.heart template=densityplot1;
    label weight="Patient Weight";
run;
```

Example 2: Density Plot and Histogram

When one or more density plots are overlaid on a histogram, the dependent axis shows the statistic indicated by the histogram's SCALE= option. The area under each density curve is equal to the area of the histogram. The following graph was generated by the "Example Program" on page 464:
Example Program

proc template;
define statgraph densityplot2;
begingraph;
  entrytitle "Patient Weight Distribution";
  entrytitle "with Fitted Normal Curve";
  entryfootnote halign=left "Framingham Heart Study";
  layout overlay;
    histogram weight / primary=true scale=count;
    densityplot weight / normal() lineattrs=graphfit;
  endlayout;
endgraph;
end;
run;
proc sgrender data=sashelp.heart template=densityplot2;
  label weight="Patient Weight";
run;

Example 3: Grouped Density Plot

The following graph was generated by the "Example Program" on page 465:
Example Program

```sas
proc template;
define statgraph densityplot;
begingraph;
entrytitle "Highway Mileage Distribution by Origin";
layout overlay /
xaxisopts=(griddisplay=on
    gridattrs=(color=lightgray pattern=dot))
yaxisopts=(griddisplay=on
    gridattrs=(color=lightgray pattern=dot));
densityplot mpg_highway / name="densityplot" group=origin;
discretelegend "densityplot" / title="Origin:";
endlayout;
endgraph;
end;
run;

proc sgrender template=densityplot data=sashelp.cars;
run;
```

DROPLINE Statement

Creates a horizontal or vertical drop line from a point to an axis.

Requirement: A DROPLINE statement must be used within a 2-D layout (for example, an OVERLAY, OVERLAYEQUATED, DATALATTICE, or DATAPANEL layout).

Syntax

```
DROPLINE X=x-axis-value | column | expression
Y=y-axis-value | column | expression </options>;
```
Summary of Optional Arguments

Appearance options

CLIP=TRUE | FALSE
  specifies whether the data for the line are considered when determining the data ranges for the axes.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of the drop line.

DATATRANSPARENCY=number
  specifies the degree of the transparency of the drop line.

LINEATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the drop line.

Axes options

DISCRETEOFFSET=number
  specifies an amount to offset all drop lines from discrete X values, or Y values, or both.

DROPTO=X | Y | BOTH
  specifies the axis to which the line is dropped.

XAXIS=X | X2
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
  specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Label options

LABEL="string" | string-column
  specifies a label for the point(s) indicated by the X= and Y= arguments.

LABELATTRS=style-element | style-element (text-options) | (text-options)
  specifies the color and font attributes of the drop line label(s).

LEGENDLABEL="string"
  specifies a label to be used in a discrete legend for this plot.

Plot reference options

NAME="string"
  assigns a name to this plot statement for reference in other template statements.

Required Arguments

X=x-axis-value | column | expression
  specifies the X coordinate of the drop line(s).

Requirement
  Values must agree in type with the X-axis data type. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Note
  When a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.
### Optional Arguments

**CLIP=TRUE | FALSE**

specifies whether the data for the line are considered when determining the data ranges for the axes.

**FALSE**

specifies that the reference line values are to be considered when the axis range is determined. The reference lines are drawn as follows based on the axis type:

- For a discrete axis, the reference line values that are not already on the axis are added to the end of the axis data list. When applicable, the axis values are then sorted:
  - If the axis values are numeric values, then they are sorted ordinally.
  - If the axis values are character values and a sorting option is applied to the axis, then they are sorted as specified by the sorting option.

Reference lines are then drawn at the specified locations.

- For a linear, log, or time axis, a new axis data list is created by performing a mathematical union of the data values and the reference line values. The reference lines are then drawn at the locations specified.

**TRUE**

specifies that the reference line values are not to be considered when the axis range is determined. The reference lines are drawn as follows based on the axis type:

- For a discrete axis, if the reference line value exactly matches a value on the axis, then a reference line is drawn at that location. Otherwise, the reference line is not drawn.

**Note:** If the axis values are formatted, then the reference line value must exactly match the formatted axis value in order for the line to be drawn.

- For a linear, log, or time axis, if the reference line value is within the axis data range, then the reference line is drawn at the specified location. Otherwise, the reference line is not drawn.
Default: FALSE

See: "boolean * on page 1479 for other Boolean values that you can use.

**DATASKIN**=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the drop line. The following figure shows drop lines with each of the skins applied.

![Diagram showing drop lines with different skins applied](image)

Default: The **DATASKIN**= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

Restriction: Starting with **SAS 9.4M1**, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the **DATASKINMAX**= option in your ODS GRAPHICS statement to increase the maximum limit.

Interaction: This option overrides the BEGINGRAPH statement **DATASKIN**= option.

**DATATRANSPARENCY**=number

specifies the degree of the transparency of the drop line.

Default: 0

Range: 0–1, where 0 is opaque and 1 is entirely transparent

Note: This option does not affect the point labels.

**DISCRETEOFFSET**=number

specifies an amount to offset all drop lines from discrete X values, or Y values, or both.

Default: 0 (no offset, all drop lines are centered on discrete X values, or discrete Y values, or both)

Range: -0.5 to +0.5 where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the right. If the Y axis is discrete, then a positive offset is up. If
REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.

Restriction
This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

Tip
Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

DROPTO=X | Y | BOTH
specifies the axis to which the line is dropped.

X
draws one or more droplines to an X axis.

Tip
The XAXIS= option determines whether the X axis or X2 axis is the endpoint for the line.

Y
draws one or more droplines to a Y axis.

Tip
The YAXIS= option determines whether the Y axis or Y2 axis is the endpoint for the line.

BOTH
draws one or more droplines to both axes.

Note
This option is valid starting in SAS 9.4M1.

Default
X

LABEL="string" | string-column
specifies a label for the point(s) indicated by the X= and Y= arguments.

Default
No label is specified

Interaction
If drawing multiple drop lines using X=column or Y=column, then you can assign corresponding labels by using a column to define the labels.

Note
Starting with SAS 9.4M1, space is reserved at the maximum end of the X axis to accommodate the length of the labels regardless of where the labels appear in the plot.

Tips
You can use the OFFSETMAX= axis option to adjust the amount of space that is reserved on the X axis for the labels.

The font and color attributes for the label are specified by the LABELATTRS= option.

LABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the drop line label(s).

Default
The GraphValueText style element.
Interaction  For this option to have any effect, the LABEL= option must also be specified.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

   “Text Options” on page 1525 for available text-options.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Default  The string specified on the NAME= option.

Restriction  This option applies only to an associated DISCRETELEGEND statement.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the drop line.

Default  The GraphReference style element.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

   “Line Options” on page 1522 for available line-options.

NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

   This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction  The string is used as the default legend label if the LEGENDLABEL= option is not used.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default  X

Restriction  Another plot that establishes a data range for the designed axis must be included.

Interaction  The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default  Y
Restriction  Another plot that establishes a data range for the designed axis must be included.

Interaction  The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details
A drop line is always drawn perpendicular from the specified point to the X (bottom), X2 (top), Y (left), or Y2 (right) axis. Axis offsets do not apply to drop lines, so they always meet the axis line.

The DROPTO= option controls whether a horizontal or vertical drop line is created. DROPTO=X specifies the horizontal axis for a vertical drop line, and DROPTO=Y specifies the vertical axis for a horizontal drop line.

A DROPLINE statement can be used only within a 2-D overlay-type layout (OVERLAY, OVERLAYEQUATED, PROTOTYPE). Another plot statement that is derived from data values that provide boundaries for the axis area must be included in the layout. For example, it can be used with a scatter plot or a histogram. You can generate multiple drop lines by specifying a column for X and Y. The column type (numeric or string) must agree with the type of data presented on the axis.

To generate both a vertical and a horizontal drop line from a single point, use multiple DROPLINE statements.

Example: DROPLINE Statement
The following graph was generated by the “Example Program” on page 472. The graph shows two DROPLINE statements originating from the same point (X=3, Y=5). One statement uses DROPTO=X and the other uses DROPTO=Y.
Example Program

```
proc template;
  define statgraph dropline;
  begingraph;
    entrytitle "Drop lines at Inflection Point";
    layout overlay / yaxisopts=(linearopts=(viewmin=0));
    seriesplot x=x y=y;
    dropline x=3 y=5 / dropto=x
      lineattrs=(color=blue pattern=dot) label="(3,5)";
    dropline x=3 y=5 / dropto=y
      lineattrs=(color=blue pattern=dot);
    endlayout;
  endgraph;
end;
run;

data test;
  do X=0 to 8 by 0.25;
    Y=(x-3)*(x-3) + 5;
    output;
  end;
run;

proc sgrender data=test template=dropline;
run;
```

ELLIPSE Statement

Creates a confidence ellipse computed from input data.

Requirements: An ELLIPSE statement must be used in a two-dimensional overlay-type layout (for example, an OVERLAY, OVERLAYEQUATED, or PROTOTYPE layout). The ELLIPSE statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area. It is typically overlaid with a scatter plot.

**Syntax**

```
ELLIPSE X=numeric-column | expression
  Y=numeric-column | expression </options>;
```

**Summary of Optional Arguments**

**Appearance options**

- `CLIP=TRUE | FALSE` specifies whether the data for the ellipse are considered when determining the data ranges for the axes.
- `DATATRANSPARENCY=number` specifies the degree of the transparency of the ellipse fill color and outline.
- `DISPLAY=(display-options) | STANDARD | ALL`
specifies which ellipse features to display.

**FILLATTRS=** *style-element | style-element (fill-options) | (fill-options)*  
specifies the appearance of the interior fill area of the ellipse.

**FILLPATTERNATTRS=** *style-element | (fill-pattern-options)*  
specifies the appearance of the pattern-filled areas.

**OUTLINEATTRS=** *style-element | style-element (line-options) | (line-options)*  
specifies the attributes of the ellipse outline.

### Axes options

**XAXIS=X | X2**  
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS=Y | Y2**  
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

### Label options

**LEGENDLABEL=** "string"  
specifies a label to be used in a discrete legend for this plot.

### Midpoint options

**GROUP=** *column | discrete-attr-var | expression*  
creates a separate ellipse for each unique group value in the specified column.

**INCLUDEMISSINGGROUP=** *TRUE | FALSE*  
specifies whether missing values of the group variable are included in the plot.

### Plot reference options

**NAME=** "string"  
assigns a name to this plot statement for reference in other template statements.

### Statistics options

**ALPHA=** *positive-number*  
sets a significance value for the confidence level to compute for the ellipse.

**FREQ=** *numeric-column | expression*  
specifies a numeric column that provides frequencies for each observation that is read.

**TYPE=** *MEAN | PREDICTED*  
specifies the type of ellipse.

### Required Arguments

**X=** *numeric-column | expression*  
specifies the numeric column for the X values.

**Y=** *numeric-column | expression*  
specifies the numeric column for the Y values.
Optional Arguments

**ALPHA=positive-number**
sets a significance value for the confidence level to compute for the ellipse.

Default: 0.05

Range: $0 < \text{number} < 1$

Note: ALPHA=0.05 represents a 95% confidence level.

See **TYPE=**

**CLIP=TRUE | FALSE**
specifies whether the data for the ellipse are considered when determining the data ranges for the axes.

**FALSE**
the data for the ellipse contribute to the data range for each axis.

  Each axis might be extended to force the display of the entire ellipse.

**TRUE**
the data for the ellipse are ignored when establishing axis scales.

  Each axis scale is determined by the other plots in the parent layout. This might result in the ellipse not being entirely displayed (clipped) if its data range is not within the data ranges of the other plots.

Default: FALSE

See "boolean" on page 1479 for other Boolean values that you can use.

**DATATRANSPARENCY=number**
specifies the degree of the transparency of the ellipse fill color and outline.

Default: 0

Range: $0–1$, where 0 is opaque and 1 is entirely transparent

Tip: The FILLATTRS= option can be used to set transparency for just the ellipse fill. You can combine this option with FILLATTRS= to set one transparency for the ellipse outline but a different transparency for the ellipse fill. Example:

  datatransparency=0.2 fillattrs=(transparency=0.6)

**DISPLAY=display-options | STANDARD | ALL**
specifies which ellipse features to display.

(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**OUTLINE**
  displays the ellipse outline

**FILL**
  displays the ellipse fill color
FILLPATTERN

displays the ellipse fill pattern

**Note** This feature applies to **SAS 9.4M5** and to later releases.

**Tip** By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the **FILLPATTERNATTRS=** option to specify a fill pattern for the ellipse.

**STANDARD**
specifies OUTLINE only

**ALL**
specifies all features: FILL, FILLPATTERN, and OUTLINE

**Default** GraphEllipse:DisplayOpts style reference.

**Tip** Use **FILLATTRS=** and **OUTLINEATTRS=** to control the appearance of the ellipse outline and fill color. Starting with **SAS 9.4M1**, use the **FILLPATTERNATTRS=** option to control the appearance of the ellipse fill pattern.

**FILLATTRS=**

**style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the interior fill area of the ellipse.

**Default** The GraphDataDefault style element.

**Interaction** For this option to have any effect, the fill must be enabled by the ODS style or the **DISPLAY=** option.

**Tip** The **DATATRANSPARENCY=** option sets the transparency for the ellipse fill and ellipse outline. You can combine this option with **DATATRANSPARENCY=** to set one transparency for the outline but a different transparency for the fill. Example:

```
DATATRANSPARENCY=0.2
FILLATTRS=(transparency=0.6)
```

**See**

"General Syntax for Attribute Options" on page 1519 for the syntax on using a **style-element**.

"Fill Color Options" on page 1520 for available **fill-options**.

**FILLPATTERNATTRS=**

**style-element | (fill-pattern-options)**
specifies the appearance of the pattern-filled areas.

**Note:** This feature applies to **SAS 9.4M5** and to later releases.

**Defaults** For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

**Interaction** For this option to take effect, the **DISPLAY=** option must include **FILLPATTERN** among the display options.
Tip
In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITIZE=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User's Guide.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Pattern Options” on page 1521 for available fill-pattern-options.

FREQ=numeric-column | expression
specifies a numeric column that provides frequencies for each observation that is read.

Default
All observations have a frequency count of 1.

Restriction
If the value of the numeric-column is missing or is less than 1, then the observation is not used in the analysis. If the value is not an integer, then only the integer portion is used.

Note
If n is the value of the numeric column for a given observation, then that observation is used n times for the purposes of any statistical computation.

GROUP=column | discrete-attr-var | expression
creates a separate ellipse for each unique group value in the specified column.

Note: This feature applies to SAS 9.4M2 and to later releases.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

A distinct ellipse is created for each group value by varying the visual attributes of the ellipse display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
</tbody>
</table>
**Display Feature**

<table>
<thead>
<tr>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill pattern 2</td>
</tr>
<tr>
<td>Outline</td>
</tr>
</tbody>
</table>

1 The default display features are determined by the DisplayOpts attribute of the GraphEllipse style element.
2 Fill patterns are valid for ellipse plots starting with SAS 9.4M5.

**Notes**
The group values are mapped in the order in which they appear in the data.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tip**
You can individually override the representations that are used to identify the groups. For example, in some ODS styles, each distinct group value is represented by a different line color and pattern. In that case, you can use the PATTERN= setting on the OUTLINEATTRS= option to assign the same line pattern to all of the ellipses.

**See**
“Example 2: 95% Confidence Ellipses by Plant Species” on page 481
“DISCRETEATTRVAR Statement” on page 1435

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**Default**
TRUE

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Tip**
The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**
“boolean ” on page 1479 for other Boolean values that you can use.
**LEGENDLABEL=**"string"
specifies a label to be used in a discrete legend for this plot.

**Default**
The *string* specified on the NAME= option.

**Restriction**
This option applies only to an associated DISCRETELEGEND statement.

**NAME=**"string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**
The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**
The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**OUTLINEATTRS=**style-element | style-element (line-options) | (line-options)
specifies the attributes of the ellipse outline.

**Defaults**
For non-grouped data, the GraphDataDefault style element.

For grouped data, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the LineThickness attribute of the GraphDataDefault style element.

**Interaction**
For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**TYPE=**MEAN | PREDICTED
specifies the type of ellipse.

**MEAN**
specifies a confidence ellipse of the mean

**PREDICTED**
specifies a prediction ellipse for a new observation

**Default**
MEAN

**See**
ALPHA= option for specifying a confidence level.

For statistical details about how the ellipse is calculated, see “Confidence and Prediction Ellipses ” on page 479.

**XAXIS=**X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
YAXIS= Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Details
Statement Description
The ELLIPSE statement can be used only within 2-D overlay-type layouts. It computes an ellipse for a set of points specified by the X and Y columns and a confidence level specified by the ALPHA= option. Use the TYPE= option to control whether a predicted or confidence ellipse is generated.

Confidence and Prediction Ellipses
Two types of ellipses can be computed for the input data (where observations correspond to points in a scatter plot). One is a confidence ellipse for the population mean (TYPE=MEAN), and the other is a prediction ellipse for a new observation (TYPE=PREDICT). Both assume a bivariate normal distribution.

Let \( \bar{Z} \) and \( S \) be the sample mean and sample covariance matrix of a random sample of size \( n \) from a bivariate normal distribution with mean \( \mu \) and covariance matrix \( \Sigma \). The variable \( \bar{Z} - \mu \) is distributed as a bivariate normal variate with mean zero and covariance \( \frac{1}{n} \Sigma \), and it is independent of \( S \). Using Hotelling’s \( T^2 \) statistic, which is defined as follows:

\[
T^2 = n(\bar{Z} - \mu)'S^{-1}(\bar{Z} - \mu)
\]

A 100(1 – \( \alpha \))% confidence ellipse for \( \mu \) is computed from the following equation:

\[
\frac{n}{n-1}(\bar{Z} - \mu)'S^{-1}(\bar{Z} - \mu) = \frac{2}{n-2}F_{2,n-2}(1-\alpha)
\]

where \( F_{2,n-2}(1-\alpha) \) is the \( (1-\alpha) \) critical value of an \( F \) distribution with degrees of freedom 2 and \( n-2 \).

A prediction ellipse is a region for predicting a new observation in the population. It also approximates a region containing a specified percentage of the population.

Denote a new observation as the bivariate random variable \( Z_{new} \). The following variable:

\[
Z_{new} - \bar{Z} = (Z_{new} - \mu) - (\bar{Z} - \mu)
\]
is distributed as a bivariate normal variate with mean zero (the zero vector) and covariance \((1 + \frac{1}{n})\Sigma\), and it is independent of \(S\). A \(100(1 - \alpha)\%\) prediction ellipse is then given by the following equation:

\[
\frac{n}{n-1}(\mathbf{Z} - \mu)'\mathbf{S}^{-1}(\mathbf{Z} - \mu) = \frac{2(n+1)}{n-2}F_{2, n-2}(1 - \alpha)
\]

The family of ellipses generated by different critical values of the \(F\) distribution has a common center (the sample mean) and common major and minor axis directions. The shape of an ellipse depends on the aspect ratio of the plot. The ellipse indicates the correlation between the two variables if the variables are standardized (by dividing the variables by their respective standard deviations). In this situation, the ratio between the major and minor axis lengths is:

\[
\sqrt{1 + r}
\]

In particular, if \(r = 0\), then the ratio is 1, which corresponds to a circular confidence contour and indicates that the variables are uncorrelated. A larger value of the ratio indicates a larger positive or negative correlation between the variables.

Examples:

Example 1: Prediction Ellipses

The following graph was generated by the “Example Program” on page 481:

![Prediction Ellipses](image)
Example Program

This example overlays two ELLIPSE statements on a SCATTERPLOT of the same data. Both ELLIPSE statements use TYPE=PREDICTED. One ELLIPSE statement uses ALPHA=0.2, and the other uses ALPHA=0.05.

```
proc template;
  define statgraph ellipse;
  begingraph;
    entrytitle "Prediction Ellipses";
    layout overlayequated / equatetype=equate;
      scatterplot x=petallength y=petalwidth /
        datatransparency=0.5;
      ellipse x=petallength y=petalwidth /
        type=predicted alpha=0.2
        name="p80" legendlabel="80%"
        outlineattrs=graphconfidence;
      ellipse x=petallength y=petalwidth /
        type=predicted alpha=0.05
        name="p95" legendlabel="95%"
        outlineattrs=graphconfidence2;
      discretelegend "p80" "p95" /
        location=inside autoalign=(topleft);
    endlayout;
    entryfootnote halign=left "Fisher's Iris Data";
  endgraph;
end;
run;
proc sgrender data=sashelp.iris template=ellipse;
run;
```

Example 2: 95% Confidence Ellipses by Plant Species

The following graph was generated by the “Example Program” on page 482:
Example Program

```sas
proc template;
   define statgraph ellipsegroup;
      begingraph;
         entrytitle "95% Prediction By Species";
         layout overlay;
            scatterplot x=petallength y=petalwidth / name="sp"
                   group=species;
            ellipse x=petallength y=petalwidth / group=species
                   type=predicted alpha=0.05 name="p95";
            mergedlegend "sp" "p95" / location=inside
                           autoalign=(topleft) across=1;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=sashelp.iris template=ellipsegroup;
run;
```

**ELLIPSEPARAM Statement**

Creates an ellipse specified by slope, axis, and origin parameters.

**Requirements:**
- An ELLIPSEPARAM statement must be used in a two-dimensional overlay-type layout (for example, an OVERLAY, OVERLAYEQUATED, or PROTOTYPE layout).
- The ELLIPSEPARAM statement must be overlaid with another plot that is derived from data values that provide boundaries for the axis area. It is typically overlaid with a scatter plot.

**Tip:**
- You can generate a single ellipse by specifying a constant for each required argument.
- You can generate multiple ellipses by specifying a numeric column for any or all required arguments. If any of the SEMIMAJOR=, SEMIMINOR=, XORIGIN= or YORIGIN= constants or columns contains a missing value, then no ellipse is drawn. To request a vertical major axis, specify SLOPE=.. (missing value) as a constant or column value.

**Syntax**

```
ELLIPSEPARAM SEMIMAJOR=number | numeric-column | expression
   SEMIMINOR=number | numeric-column | expression
   SLOPE=number | numeric-column | expression
   XORIGIN=number | numeric-column | expression
   YORIGIN=number | numeric-column | expression <options>;
```

**Summary of Optional Arguments**

**Appearance options**

```
CLIP=TRUE | FALSE
```

specifies whether the data for the ellipse are considered when determining the data ranges for the axes.
DATATRANSPARENCY=number
specifies the degree of the transparency of the ellipse fill color and outline.

DISPLAY=(display-options) | STANDARD | ALL
specifies which ellipse features to display.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the interior fill area of the ellipse.

FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

INDEX=positive-integer-column | expression
specifies indices for mapping ellipse attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the ellipse outline.

Axes options
XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Label options
LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options
GROUP=column | discrete-attr-var | expression
creates a separate ellipse for each unique group value of the specified column.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments
SEMIMAJOR=number | numeric-column | expression
specifies half the length of the major axis for the ellipse. Values are in the same units as the data. The SEMIMAJOR value can be greater than, smaller than, or equal to the SEMIMINOR value.

Restriction
If a numeric column is specified and the column contains negative values, then no ellipses are drawn. In SAS 9.4M2 and in earlier releases, the same is true if the column contains missing values. Starting with SAS 9.4M3, if the column contains a missing value, the ellipse for that observation is not drawn. However, the remaining ellipses that have nonmissing values for this argument and for SEMIMINOR= are drawn.
SEMIMINOR=number | numeric-column | expression
specifies half the length of the minor axis for the ellipse. Values are in the same units as the data. The SEMIMINOR value can be greater than, smaller than, or equal to the SEMIMAJOR value.

Restriction  If a numeric column is specified and the column contains negative values, then no ellipses are drawn. In SAS 9.4M3 and in earlier releases, the same is true if the column contains missing values. Starting with SAS 9.4M3, if the column contains a missing value, the ellipse for that observation is not drawn. However, the remaining ellipses that have nonmissing values for this argument and for SEMIMAJOR= are drawn.

SLOPE=number | numeric-column | expression
specifies the slope of the major axis for the ellipse. Slope can be positive or negative.

Note  The slope value is in the data space and might or might not be maintained in the screen space. Thus, setting SLOPE=1 does not always generate a 45-degree line on the screen.

Tip  Setting SLOPE=0 creates a major axis parallel to the X axis. Setting SLOPE=. (missing value) creates a major axis parallel to the Y axis.

XORIGIN=number | numeric-column | expression
specifies the X coordinate of the center of the ellipse. Values are in the units of the data.

Tip  By default, if the value specified for the XORIGIN= option is outside of the X-axis data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

YORIGIN=number | numeric-column | expression
specifies the Y coordinate of the center of the ellipse. Values are in the units of the data.

Tip  By default, if the value specified for the YORIGIN= option is outside of the Y-axis data range, then the data range is extended to include the specified point. This behavior can be changed with the CLIP= option.

Optional Arguments

CLIP=TRUE | FALSE
specifies whether the data for the ellipse are considered when determining the data ranges for the axes.

FALSE  The data for the ellipse contribute to the data range for each axis.

Each axis might be extended to force the display of the entire ellipse.

TRUE  The data for the ellipse are ignored when establishing axis scales.

Each axis scale is determined by the other plots in the parent layout. This might result in the ellipse not being entirely displayed (clipped) if its data range is not within the data ranges of the other plots.

Default  FALSE
See “boolean” on page 1479 for other Boolean values that you can use.

**DATATRANSPIRACY=** *number*

specifies the degree of the transparency of the ellipse fill color and outline.

**Default** 0

**Range** 0–1, where 0 is opaque and 1 is entirely transparent

**Tip** The **FILLATTRS**= option can be used to set transparency for just the ellipse fill. You can combine this option with **FILLATTRS**= to set one transparency for the ellipse outline but a different transparency for the ellipse fill. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**DISPLAY=(** *display-options*) | STANDARD | ALL

specifies which ellipse features to display.

**display-options**

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**OUTLINE**

displays the ellipse outline

**FILL**

displays a filled ellipse

**FILLPATTERN**

displays the ellipse fill pattern

**Note** This feature applies to **SAS 9.4M5** and to later releases.

**Tip** By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the **FILLPATTERNATTRS**= option to specify a fill pattern for the ellipse.

**STANDARD**

displays OUTLINE only

**ALL**

displays all features: FILL, FILLPATTERN, and OUTLINE

**Default** GraphEllipse:DisplayOpts style reference.

**Tip** Use **FILLATTRS**= and **OUTLINEATTRS**= to control the appearance of the ellipse outline and fill color. Starting with **SAS 9.4M5**, use the **FILLPATTERNATTRS**= on page 486 option to control the appearance of the fill pattern.

**FILLATTRS=** *style-element | style-element (fill-options) | (fill-options)*

specifies the appearance of the interior fill area of the ellipse.

**Defaults** For non-grouped data, the GraphDataDefault style element.

For grouped data, the GraphData1–GraphDataN style elements.
**Interaction**

For this option to have any effect, the fill must be enabled by the ODS style or the `DISPLAY=` option.

---

**Tip**

The `DATATRANSPARENCY` option sets the transparency for the ellipse fill and ellipse outline. You can combine this option with `DATATRANSPARENCY=` to set one transparency for the outline but a different transparency for the fill. Example:

```
DATATRANSPARENCY=0.2 fillattrs=(transparency=0.6)
```

---

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Fill Color Options” on page 1520 for available `fill-options`.

---

**FILLPATTERNATTRS=**

`style-element | (fill-pattern-options)`

specifies the appearance of the pattern-filled areas.

---

**Note:** This feature applies to SAS 9.4M5 and to later releases.

---

**Defaults**

For non-grouped data, the ContrastColor and FillPattern attributes of `GraphDataDefault`.

For grouped data, the ContrastColor and FillPattern attributes of `GraphData1–GraphDataN`.

---

**Interaction**

For this option to take effect, the `DISPLAY=` option must include `FILLPATTERN` among the display options.

---

**Tip**

In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify `ATTRPRIORITY=NONE` in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in *SAS Graph Template Language: User’s Guide*.

---

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Fill Pattern Options” on page 1521 for available `fill-pattern-options`.

---

**GROUP=**

`column | discrete-attr-var | expression`

creates a separate ellipse for each unique group value of the specified column.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a `DISCRETEATTRVAR` statement.

**Restriction**

A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

---

A distinct ellipse is created for each group value by varying the visual attributes of the ellipse display features. The display features are controlled by the current ODS style or by the `DISPLAY=` option. The default group appearance for each display feature is shown in the following table.
<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern. If the active ODS style does not specify FillPattern, fill patterns are not displayed by default.</td>
</tr>
<tr>
<td>Outline</td>
<td>ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
</tbody>
</table>

1. The default display features are determined by the DisplayOpts attribute of the GraphEllipse style element.
2. Fill patterns are valid for ellipse plots starting with SAS 9.4M5.

The representations that are used to identify the groups can be overridden individually. For example, each distinct group value might be represented by a different line pattern for the ellipses, but the PATTERN= suboption of the OUTLINEATTRS= option could be used to assign the same line pattern to all ellipse outlines, letting outline color distinguish group values.

Defaults

If ellipse fills are enabled by the ODS style or by the DISPLAY= option, then each distinct group value is represented in the plot by a different fill color. Starting with SAS 9.4M5, each distinct group value can be represented by a different fill pattern, with or without a different fill color. The fill color and fill pattern are defined by the Color and FillPattern attributes of the GraphData1–GraphDataN and GraphMissing style elements.

If only ellipse outlines are enabled by the ODS style or by the DISPLAY= option, then each distinct group value is represented in the plot by a different ellipse outline color and line style. The outline colors and line styles are defined by the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements.

Restriction

When the GROUP= option is specified, the group value must be a character or numeric column. For each group value, there must be a numeric column that does not contain missing values for SEMIMAJOR=, SEMIMINOR=, XORIGIN=, and YORIGIN=. The SLOPE= column can contain missing values. Under these circumstances, an ellipse is drawn for each group value.

Interaction

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

Note

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default</th>
<th>Interaction</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDEMISSINGGROUP=TRUE</td>
<td>specifies whether missing values of the group variable are included in the plot.</td>
<td>TRUE</td>
<td>For this option to take effect, the GROUP= option must also be specified.</td>
<td>The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element. See “boolean” on page 1479 for other Boolean values that you can use.</td>
</tr>
<tr>
<td>INDEX=positive-integer-column</td>
<td>specifies indices for mapping ellipse attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.</td>
<td></td>
<td>For this option to take effect, the GROUP= option must also be specified.</td>
<td>The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use. If you do not use this option, then the group values are mapped in the order of the data. You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.</td>
</tr>
<tr>
<td>LEGENDLABEL=&quot;string&quot;</td>
<td>specifies a label to be used in a discrete legend for this plot.</td>
<td>The string specified on the NAME= option.</td>
<td>This option applies only to an associated DISCRETELEGEND statement.</td>
<td>If the GROUP= option is specified, then this option is ignored.</td>
</tr>
</tbody>
</table>
NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction The string is used as the default legend label if the LEGENDLABEL= option is not used.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the ellipse outline.

Defaults For non-grouped data, the GraphDataDefault style element.
          For grouped data, the GraphData1–GraphDataN style elements.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Restriction Another plot that establishes a data range for the designed axis must be included.

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default Y

Restriction Another plot that establishes a data range for the designed axis must be included.

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details
The ELLIPSEPARM statement plots an ellipse with specified semimajor and semiminor axis lengths, a specified slope for the major axis, and the ellipse center. The ellipse center is the point of intersection of the semimajor and semiminor axes. It can be used only within a two-dimensional overlay-type layout (for example, an
OVERLAY, OVERLAYEQUATED, or PROTOTYPE layout). The ELLIPSEPARM statement does not perform computations on input data to derive the location and shape of the ellipse. Another plot that is derived from data values that provide boundaries for the axis area must be included in the layout. It is typically overlaid with a scatter plot. See “ELLIPSE Statement” on page 472 for information about computed ellipses.

Example: ELLIPSEPARM Statement

The following graph was generated by the “Example Program” on page 491:

Overview

This is a simplified version of the CorrLoadPlot template for PROC PLS in the SAS/STAT product. It consists of overlaid scatter plots of the scores of the first two factors, the loadings of the model effects, and the loadings of the dependent variables. The loadings are scaled so that the amount of variation in the variables that is explained by the model is proportional to the distance from the origin; circles indicating various levels of explained variation are also overlaid.

The circles are drawn with ELLIPSEPARM statements by setting the SEMIMAJOR and SEMIMINOR lengths to be the same. Notice that all circles are concentric because they share the same origin (0,0). The radius of each circle is passed dynamically.

Also note that an OVERLAYEQUATED layout was used to force the length of unit intervals on both axes to be the same.

The input data shown is representative of that computed by PROC PLS for the Correlation Loadings Plot. For more details, see the first example for PROC PLS in the SAS/STAT user’s guide.
Example Program

Here is the SAS program for this example.

```sas
proc template;
define statgraph ellipseparm;
dynamic RADIUS1 RADIUS2 RADIUS3 RADIUS4;
begingraph;
entrytitle "Correlation Loading Plot";
layout overlayequated / equatetype=square
commonaxisopts=
tickvaluelist=(-1 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 1)
viewmin=-1 viewmax=1
xaxisopts=
(label="Factor 1" offsetmin=0.05 offsetmax=0.05)
yaxisopts=
(label="Factor 2" offsetmin=0.05 offsetmax=0.05);
ellipseparm semimajor=RADIUS1 semiminor=RADIUS1 slope=0
xorigin=0 yorigin=0 / clip=true display=(outline)
outlineattrs=(pattern=dash) datatransparency=0.75;
ellipseparm semimajor=RADIUS2 semiminor=RADIUS2 slope=0
xorigin=0 yorigin=0 / clip=true display=(outline)
outlineattrs=(pattern=dash) datatransparency=0.75;
ellipseparm semimajor=RADIUS3 semiminor=RADIUS3 slope=0
xorigin=0 yorigin=0 / clip=true display=(outline)
outlineattrs=(pattern=dash) datatransparency=0.75;
ellipseparm semimajor=RADIUS4 semiminor=RADIUS4 slope=0
xorigin=0 yorigin=0 / clip=true display=(outline)
outlineattrs=(pattern=dash) datatransparency=0.75;
scatterplot x=xcirclelabel y=ycirclelabel / primary=true
markercharacter=circlelabel datatransparency=0.75;
scatterplot x=corr1 y=corr2 / name="ScatterVars"
group=corrgroup markercharacter=corrlabel;
discretelegend "ScatterVars";
endlayout;
endGraph;
end;
run;

data corrplot;
infile cards missover dlm="#"
length CorrGroup $ 18 CorrLabel CircleLabel $ 8;
input Corr1 Corr2 CorrGroup CorrLabel xCircleLabel
ycircleLabel CircleLabel;
datalines;
-0.179 -0.268 # Predictor Loading # S1 # 0 # 0.5 # 25%
0.105 # 0.332 # Predictor Loading # S2 # 0 # -0.5 # 25%
-0.654 # 0.094 # Predictor Loading # S3 # 0 # 0.707 # 50%
-0.653 # 0.685 # Predictor Loading # S4 # 0 # -0.707 # 50%
0.096 # 0.059 # Predictor Loading # S5 # 0 # 0.866 # 75%
0.132 # 0.036 # Predictor Loading # L1 # 0 # -0.866 # 75%
0.087 # 0.156 # Predictor Loading # L2 # 0 # 1 # 100%
0.940 # 0.160 # Predictor Loading # L3 # 0 # -1 # 100%
0.607 -0.350 # Predictor Loading # L4
0.096 # 0.059 # Predictor Loading # L5
```
-0.111 # -0.534 # Predictor Loading # P1
0.003 # 0.256 # Predictor Loading # P2
0.293 # 0.551 # Predictor Loading # P3
-0.480 # 0.643 # Predictor Loading # P4
-0.096 # -0.059 # Predictor Loading # P5
0.946 # 0.279 # Response Loading # log_RAI
-0.196 # 0.403 # Observation # 1
0.020 # -0.001 # Observation # 2
-0.195 # 0.324 # Observation # 3
0.021 # -0.079 # Observation # 4
-0.009 # -0.274 # Observation # 5
0.567 # 0.294 # Observation # 6
-0.096 # -0.059 # Observation # 7
0.258 # 0.210 # Observation # 8
-0.104 # -0.309 # Observation # 9
-0.187 # -0.458 # Observation # 10
0.051 # -0.078 # Observation # 11
0.017 # -0.260 # Observation # 12
-0.621 # 0.372 # Observation # 13
0.392 # 0.138 # Observation # 14
0.080 # -0.221 # Observation # 15
;
run;

proc sgrender data=corrplot template=ellipseparm;
  dynamic radius1=0.50 radius2=0.71 radius3=0.87 radius4=1;
run;

FRINGE PLOT Statement

Creates a fringe plot on the X axis of an X-Y plot.

Syntax

FRINGE PLOT numeric-column | expression </options>;

Summary of Optional Arguments

Appearance options

DATATRANSPARENCY=number
specifies the degree of the transparency of the fringe.

FRINGEHEIGHT=dimension
specifies the height of the fringe lines.

INDEX=positive-integer-column | expression
specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the fringe lines for the data points.

Axes options
X AXIS = X | X2
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Data tip options
  ROLENAME = (role-name-list)
    specifies user-defined roles that can be used to display information in the data tips.
  TIP = (role-list) | NONE
    specifies the information to display when the cursor is positioned over a fringe line.
  TIPFORMAT = (role-format-list)
    specifies display formats for tip columns.
  TIPLABEL = (role-label-list)
    specifies display labels for tip columns.

Label options
  LEGENDLABEL = "string"
    specifies a label to be used in a discrete legend for this plot.

Midpoint options
  GROUP = column | discrete-attr-var | expression
    creates a distinct set of lines for each unique group value of the specified column.
  INCLUDEMISSINGGROUP = TRUE | FALSE
    specifies whether missing values of the group variable are included in the plot.

Plot reference options
  NAME = "string"
    assigns a name to this plot statement for reference in other template statements.

Required Argument
  numeric-column | expression
    specifies a column that provides the X coordinates of the data values.

Optional Arguments
  DATATRANSPARENCY = number
    specifies the degree of the transparency of the fringe.
    Default 0
    Range 0–1, where 0 is opaque and 1 is entirely transparent
  FRINGEHEIGHT = dimension
    specifies the height of the fringe lines.
    Default 10 px
    See "dimension" on page 1480
GROUP=column | discrete-attr-var | expression
creates a distinct set of lines for each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default Each distinct group value is represented in the plot by a different color. The colors are determined by the ContrastColor attribute of the GraphData1–GraphDataN and GraphMissing style elements. The line style and line thickness are determined by the GraphDataDefault style element.

Interactions The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of colors and line patterns.

The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Notes You can override the representations that are used to identify the groups. For example, you can use the LINEATTRS=(PATTERN=pattern) option to assign the same line pattern to all of the plot’s line patterns, letting line color indicate group values. Likewise, you can use LINEATTRS=(COLOR=color) to assign the same color to all lines, letting line pattern indicate group values.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

See "DISCRETEATTRVAR Statement" on page 1435

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Default TRUE

Interaction For this option to take effect, the GROUP= option must also be specified.

Tip The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.
See “boolean” on page 1479 for other Boolean values that you can use.

**INDEX=**positive-integer-column | expression

specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements**

The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**

For this option to take effect, the GROUP= option must also be specified.

**Notes**

The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip**

You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL=“string”**

specifies a label to be used in a discrete legend for this plot.

**Default**

The X-column label. If a label is not defined, then the X-column name is used.

**Restriction**

This option applies only to an associated DISCRETELEGEND statement.

**LINEATTRS=**style-element | style-element (line-options) | (line-options)

specifies the attributes of the fringe lines for the data points.

**Default**

The GraphDataDefault style element.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**NAME=“string”**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**

The string is case sensitive, cannot contain spaces, and must define a unique name within the template.
This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction

The string is used as the default legend label if the LEGENDLABEL= option is not used.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

*(role-name-list)*

a space-separated list of role-name = column pairs.

**Example**

The following example assigns the column Obs to the user-defined role TIP:

```
ROLENAME=(TIP1=OBS)
```

**Default**

No user-defined roles

**Requirement**

The role names that you choose must be unique and different from the predefined role X.

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over a fringe line. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the fringe plot can be specified along with roles that do.

*role-list*

an ordered, space-separated list of unique FRINGEPLOT and user-defined roles.

The FRINGEPLOT role is X.

**Tip**

User-defined roles are defined with the ROLENAME= option.

**Example**

The following example displays data tips for the columns assigned to the roles X and Y, as well as the column Pct, which is not assigned to any pre-defined FRINGEPLOT role. The Pct column should appear first in the data tip:

```
ROLENAME=(TIP1=PCT)
TIP=(X TIP1)
```

**NONE**

suppresses data tips from the plot.

**Default**

The column assigned to the X role is automatically included in the data tip information.

**Requirement**

To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interaction**

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is
specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip

The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example

ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default

The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction

Only the roles that appear in the TIP= option are used.

Requirement

A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list

a space-separated list of rolename ="string" pairs.

Example

ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

Default

The column label or column name of the column assigned to the role.

Restriction

Only the roles that appear in the TIP= option are used.

Requirement

A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default

X

Interaction

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Note

The display of the fringe lines is always anchored on the X axis (at the bottom of the plot area), even when the FRINGEPLOT’s X-column data range is mapped to the X2 axis.
Details

In a FRINGE PLOT, each fringe line represents the location of the corresponding raw data value on the X axis. All fringe lines are of equal length.

Example: FRINGE PLOT Statement

The following graph was generated by the “Example Program” on page 498:

```
proc template;
  define statgraph fringeplot;
  dynamic VAR VARLABEL;
  begingraph;
    entrytitle "Histogram and Fringeplot";
    layout overlay / xaxisopts=(label=VARLABEL)
                  yaxisopts=(offsetmin=0.03);
    fringeplot VAR / datatransparency=0.75
                  fringeheight=3pct;
    histogram VAR;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template=fringeplot;
  dynamic var="weight" varlabel="Weight (LBS)";
run;
```
HEATMAP Statement

Creates a plot of color-coded rectangles for the response variable of a pair of X and Y variables after it bins the data in two dimensions.

Note: This feature applies to SAS 9.4M3 and to later releases.

Syntax

```
HEATMAP X=numeric-column | expression
       Y=numeric-column | expression </options>;
```

Summary of Optional Arguments

Appearance options

- **COLORMODEL=** `color-ramp-style-element | (color-list)`
  specifies a color ramp to use with the COLORRESPONSE= option.
- **COLORRESPONSE=** `numeric-column | range-attr-var | expression`
  specifies a numeric column that is used to color the heat map regions.
- **COLORSTAT=** `FREQ | PCT | PROPORTION | SUM | MEAN`
  specifies the statistic to be calculated for the COLORRESPONSE= column.
- **DATATRANSPARENCY=** `number`
  specifies the degree of the transparency of the outline and fill for each region.
- **DISPLAY=** `(display-options) | STANDARD | ALL`
  specifies which region features to display.
- **FILLATTRS=** `(TRANSPARENCY=number)`
  specifies the transparency of the interior fill area of the regions.
- **OUTLINEATTRS=** `style-element | style-element (line-options) | (line-options)`
  specifies the appearance of the rectangle outlines.
- **REVERSECOLORMODEL=** `TRUE | FALSE`
  specifies whether to reverse the gradient that is specified by the ODS style that is in effect or by the COLORMODEL= option.
- **WEIGHT=** `numeric-column`
  specifies a variable in the input data set that contains values to be used as a priori weights for the colored-region calculations.
- **XGAP=** `number`
  specifies the amount of horizontal space on either side of each color-coded region in the heat map.
- **YGAP=** `number`
  specifies the amount of horizontal space on either side of each color-coded region in the heat map.

Axes options

- **PRIMARY=** `TRUE | FALSE`
  specifies that the data columns for this plot are used for determining default axis features.
XAXIS=X | X2
specifies whether data is mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

XBINAXIS=TRUE | FALSE
specifies whether to use bins as the basis for x-axis tick marks.

XENDLABELS=TRUE | FALSE
specifies whether the X-axis ticks and tick values are placed at the bin end-points or at the bin mid-points.

YAXIS=Y | Y2
specifies whether data is mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

YBINAXIS=TRUE | FALSE
specifies whether to use bins as the basis for x-axis tick marks.

YENDLABELS=TRUE | FALSE
specifies whether the Y-axis ticks and tick values are placed at the bin end-points or at the bin mid-points.

**Binning options**

NXBINS=positive-integer
specifies the number of bins to use for the X role.

NYBINS=positive-integer
specifies the number of bins to use for the Y role.

XBINSIZE=positive-number
specifies the size of bins along the X role, in data units.

XBINSTART=number
specifies the data value for the first bin of the X role.

XBOUNDARY=UPPER | LOWER
specifies how an input value is counted when it lies on the endpoint of an X bin.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the XBINSTART= value represents the midpoint, lower endpoint, or upper endpoint of the bin.

YBINSIZE=positive-number
specifies the size of bins along the Y role, in data units.

YINSTART=number
specifies the data value for the first bin of the Y role.

YBOUNDARY=UPPER | LOWER
specifies how an input value is counted when it lies on the endpoint of an Y bin.

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the YINSTART= value represents the midpoint, lower endpoint, or upper endpoint of the bin.

**Data option**

DISCRETEX=TRUE | FALSE
specifies whether the X axis is discrete when X= specifies a numeric column.

DISCRETEY=TRUE | FALSE
specifies whether the Y axis is discrete when Y= specifies a numeric column.

**Data tip options**
TIP=(role-list) | NONE
    specifies the information to display when the cursor is positioned over a rectangle.
TIPFORMAT=(role-format-list)
    specifies display formats for tip columns.
TIPLABEL=(role-label-list)
    specifies display labels for tip columns.

Plot reference options
NAME="string"
    assigns a name to a HEATMAP statement for reference in other template statements and log messages.

Statistics options
FREQ=numeric-column | expression
    specifies a column for the frequency count for each observation in the input data.

Required Arguments
X=column | expression
    specifies the column for the X values.

Requirement
If X= specifies a numeric column, then the DISCRETEX= option must be set correctly for the X-axis type. When it is not set correctly, the heat map is not drawn. If the X-axis type is discrete, then you must specify DISCRETEX=TRUE in the HEATMAP statement. Otherwise, DISCRETEX=FALSE must be in effect.

See DISCRETEX= on page 503

Y=column | expression
    specifies the column for the Y values.

Requirement
If Y= specifies a numeric column, then the DISCRETEY= option must be set correctly for the Y-axis type. When it is not set correctly, the heat map is not drawn. If the Y-axis type is discrete, then you must specify DISCRETEY=TRUE in the HEATMAP statement. Otherwise, DISCRETEY=FALSE must be in effect.

See DISCRETEY= on page 503

Optional Arguments
COLORMODEL=color-ramp-style-element | (color-list)
    specifies a color ramp to use with the COLORRESPONSE= option.

color-ramp-style-element
    specifies the name of a color-ramp style element.

The style element should contain these style attributes:
STARTCOLOR
    specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR
specifies the color for the midpoint of the range of the COLORRESPONSE= column.

ENDCOLOR
specifies the color for the highest data value of the COLORRESPONSE= column.

(color-list)
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

See “color ” on page 1480

Default The ThreeColorRamp style element

Interaction For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

COLORRESPONSE= numeric-column | range-attr-var | expression
specifies a numeric column that is used to color the heat map regions.

range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“RANGEATTRVAR Statement”

Restriction A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Tips To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

Use the COLORSTAT= option to specify the statistic to compute for the COLORRESPONSE= column.

COLORSTAT= FREQ | PCT | PROPORTION | SUM | MEAN
specifies the statistic to be calculated for the COLORRESPONSE= column.

Default FREQ

Interaction When FREQ, PCT, or PROPORTION is specified, the colors are controlled by the FREQ= and WEIGHT= option values. When SUM
or MEAN is specified, the colors are controlled by the FREQ=, WEIGHT=, and COLORRESPONSE= option values.

**See**

- COLORRESPONSE=
- FREQ=
- WEIGHT=

**DATATRANSPARENCY=number**

specifies the degree of the transparency of the outline and fill for each region.

**Default**

0

**Range**

0–1, where 0 is opaque and 1 is entirely transparent

**Tip**

Use the FILLATTRS= option to set transparency for just the rectangle fills. You can combine this option with FILLATTRS= to set one transparency for the rectangle outlines and a different transparency for the rectangle fills. Example:

`datatransparency=0.2 fillattrs=(transparency=0.6)`

**DISCRETEX=TRUE | FALSE**

specifies whether the X axis is discrete when X= specifies a numeric column.

**Default**

FALSE

**Requirement**

If X= specifies a numeric column and the X-axis type is discrete, then you must specify DISCRETEX=TRUE. Otherwise, the heat map might not be drawn.

**Interactions**

If X= specifies a character column, then this option is ignored, and the X axis is considered to be discrete.

When this option is set to TRUE, the XENDLABELS= option is ignored.

**See**

- X= on page 501

**DISCRETEY=TRUE | FALSE**

specifies whether the Y axis is discrete when Y= specifies a numeric column.

**Default**

FALSE

**Requirement**

If Y= specifies a numeric column and the Y-axis type is discrete, then you must specify DISCRETEY=TRUE. Otherwise, the heat map might not be drawn.

**Interactions**

If Y= specifies a character column, then this option is ignored, and the Y axis is considered to be discrete.

When this option is set to TRUE, the YENDLABELS= option is ignored.

**See**

- Y= on page 501

**DISPLAY=(display-options) | STANDARD | ALL**

specifies which region features to display.
**display-options**

A space-separated list of options, enclosed in parentheses.

Currently, only **OUTLINE** is supported, which displays outlined, filled rectangles.

**STANDARD**

Displays the region fill color

**ALL**

Displays the region fill color and outline

Default: **STANDARD**

**FILLATTRS=(TRANSAPRENCY=number)**

Specifies the transparency of the interior fill area of the regions.

Default: The **DATATRANSPARENCY=** option value

Range: 0–1, where 0 is opaque and 1 is entirely transparent

Restriction: Only the **TRANSAPRENCY=** sub option is honored. If a style element or any other fill sub option is specified, it is ignored by the **HEATMAP** statement.

Interaction: This option overrides the **DATATRANSPARENCY=** option only for the colored region. It does not affect the region outlines.

Tip: Use this option when you want the region fill to have a different degree of transparency than the outline.

**FREQ=numeric-column | expression**

Specifies a column for the frequency count for each observation in the input data.

Restrictions: If the value of **FREQ** for a given observation is missing or is less than 1, that observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

**NAME="string"**

Assigns a name to a **HEATMAP** statement for reference in other template statements and log messages.

Restriction: The **string** is case sensitive, cannot contain spaces, and must define a unique name within the template.

Tip: This option is used mostly with legend statements in order to coordinate the use of colors and line patterns between the graph and the legend.

**NXBINS=positive-integer**

Specifies the number of bins to use for the X role. The system determines the **XBINSIZE=** and **XBINSTART=** values if they are not specified. The bins always span the range of the data.

Default: Determined by the system.

Restriction: The number of bins is the product of the X and Y bins. Starting with **SAS 9.4M4**, the number of bins is limited to 100,000, by default.
Requirement: You must specify a value of 2 or greater. Otherwise, this option is ignored.

Tip: Starting with SAS 9.4M4, the limit on the number of bins is specified by the ODS GRAPHICS statement NXYBINSMAX= option, which defaults to 100,000. See NXYBINSMAX= on page 1494.

NYBINS=positive-integer
specifies the number of bins to use for the Y role. The system determines the YBINSIZE= and YBINSTART= values if they are not specified. The bins always span the range of the data.

Default: Determined by the system.

Restriction: The number of bins is the product of the X and Y bins. Starting with SAS 9.4M4, the number of bins is limited to 100,000, by default.

Requirement: You must specify a value of 2 or greater. Otherwise, this option is ignored.

Tip: Starting with SAS 9.4M4, the limit on the number of bins is specified by the ODS GRAPHICS statement NXYBINSMAX= option, which defaults to 100,000. See NXYBINSMAX= on page 1494.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the rectangle outlines.

Defaults: For filled regions, the GraphOutlines style element
For unfilled regions, the GraphOutlinesUnfilled style element

Interaction: For this option to have any effect, outlines must be enabled by the ODS style or by the DISPLAY= option.

Note: When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

See “General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element value
“Line Options” on page 1522 for available line-options values

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot are used for determining default axis features.

Default: FALSE. If no plot in an OVERLAY layout is designated as primary, the data columns associated with the first plot that could be primary is considered primary on a per-axis basis.

Restrictions: Only one plot in an overlay can be primary on a per-axis basis. If multiple plots specify PRIMARY=TRUE for the same axis, the last one specified is considered the primary plot.
This option is ignored if the plot is placed in a GRIDDED or LATTICE layout block.

**Tip**

This option is needed only when there are two or more plots within an overlay that contribute to a common axis. If PRIMARY=TRUE for one of them, then that plot's data columns are used to determine the axis features, regardless of where this plot statement occurs within the OVERLAY layout block.

**REVERSECOLORMODEL=TRUE | FALSE**

specifies whether to reverse the gradient that is specified by the ODS style that is in effect or by the COLORMODEL= option.

*Default*  FALSE

*See*  COLORMODEL= on page 501

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over a rectangle. If this option is used, the specified information replaces all the information that is displayed by default. You can specify roles for columns that do not contribute to the bar chart along with roles that do.

*(role-list)*

an ordered, space-separated list of unique HEATMAP roles and user-defined roles.

HEATMAP roles include X, Y, and the implicit count.

*Example*  To display data tips for the columns assigned to the roles X and Y as well as the user-defined role TIP1:

ROLENAMESPACE=(TIP1=OBS)
TIP=(TIP1 X Y)

**NONE**

suppresses data tips and URLs (if requested) in the graph output.

*Default*  The columns assigned to these roles are automatically included in the data tip information: X and Y

*Requirement*  To generate data tips, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and write the graphs to the ODS HTML destination.

**Tip**  You can control the labels and formats for the TIP variables with the TIPLABEL= and TIPFORMAT= options.

**TIPFORMAT=(role-format-list)**

specifies display formats for tip columns.

*(role-format-list)*

a space-separated list of role-name = format pairs.

*Example*  To specify a format for the user-defined TIP1 role:

ROLENAMESPACE=(TIP1=PCT)
TIP=(TIP1 X Y)
TIPFORMAT=(TIP1=PERCENT7.2)
The column format of the variable assigned to the role or BEST6. if no format is assigned to a numeric column.

**Requirement**
This option provides a way to control the formats of columns that appear in data tips. Only the roles that appear in the `TIP=` option are used.

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns.

*(role-label-list)*
a space-separated list of `role-name ="string"` pairs.

**Example**
To specify a label for the user-defined TIP1 role:

```plaintext
ROLENAME=(TIP1=PCT)
TIP=(TIP1 X Y)
TIPLABEL=(TIP1="Percent")
```

The column label or column name of the variable assigned to the role.

**Requirement**
This option provides a way to control the labels of columns that appear in data tips. Only the roles that appear in the `TIP=` option are used.

**WEIGHT=numeric-column**
specifies a variable in the input data set that contains values to be used as a priori weights for the colored-region calculations.

**Requirement**
The values of the weight variable must be nonnegative.

**Note**
If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

**XAXIS=X | X2**
specifies whether data is mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default**
X

**Interaction**
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**XBINAXIS=TRUE | FALSE**
specifies whether to use bins as the basis for x-axis tick marks. When this option is FALSE, a standard axis is used, ignoring bin boundaries and midpoints.

**Default**
TRUE

**Interactions**
When this option is TRUE, the `XENDLABELS=` option determines how the axis ticks and value labels are displayed. When this option is FALSE, the `XENDLABELS=` option is ignored.

When this option is TRUE, the system attempts to compute nice values for the bin end points. In that case, the sum of the bin data ranges might exceed the actual data range.
When this option is TRUE, the axis ticks are in predetermined locations and are not changed when the parent layout specifies axis suboptions such as TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

See “boolean” on page 1479 for other Boolean values that you can use.

**XBINSIZE=positive-number**

specifies the size of bins along the X role, in data units. The system determines the **NXBINS=** and **XBINSTART=** values if they are not specified. The bins always span the X data range.

Default: Determined by the system.

Restriction: Starting with SAS 9.4M1, the number of bins is limited to approximately 100,000. Starting with SAS 9.4M4, the limit is specified by the ODS GRAPHICS statement option NXYBINSMAX=, which defaults to 100,000. If the total number of bins that are computed from the data, the XBINSTART= value, and the XBINSIZE= value exceeds the limit, SAS computes a new bin-size value that yields approximately the maximum number of bins. A warning of the change is written to the SAS log in that case.

Interaction: The XGAP= option is applied after this option.

Tip: Starting with SAS 9.4M4, you can use the ODS GRAPHICS statement NXYBINSMAX= option to change the limit on the number of bins. See NXYBINSMAX= on page 1494.

**XBINSTART=number**

specifies the data value for the first bin of the X role. The system determines the **NXBINS=** and **XBINSIZE=** values if they are not specified. The bins always span the X data range.

Default: Determined by the system.

Interaction: The XVALUES= option specifies how this value is interpreted.

**XBOUNDARY=UPPER | LOWER**

specifies how an input value is counted when it lies on the endpoint of an X bin. If this option is set to UPPER, then the value is counted as one of the values in the upper bin. Otherwise, it is counted in the lower bin.

Default: UPPER

**XENDLABELS=TRUE | FALSE**

specifies whether the X-axis ticks and tick values are placed at the bin endpoints or at the bin mid-points.

Default: FALSE. The axis ticks and tick values are placed at the bin mid-points.

Restriction: This option is ignored when the X axis is discrete.

Interaction: This option is ignored when XBINAXIS=FALSE is in effect.

Tip: The axis ticks and tick value placements are independent of the XVALUES= option.
XGAP=number
specifies the amount of horizontal space on either side of each color-coded region in the heat map.

Default 0
Interaction If the XBINSIZE= option is specified, the XGAP= value is subtracted from the specified X bin size.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the XBINSTART= value represents the midpoint, lower endpoint, or upper endpoint of the bin.

Default MIDPOINTS
See XBINSTART= on page 508
XENDLABELS= on page 508

YAXIS=Y | Y2
specifies whether data is mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default Y
Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YBINAXIS=TRUE | FALSE
specifies whether to use bins as the basis for x-axis tick marks. When this option is FALSE, a standard axis is used, ignoring bin boundaries and midpoints.

Default TRUE
Interactions When this option is TRUE, the YENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is FALSE, the YENDLABELS= option is ignored.

When this option is TRUE, the system attempts to compute nice values for the bin end points. In that case, the sum of the bin data ranges might exceed the actual data range.

When this option is TRUE, the axis ticks are in predetermined locations and are not changed when the parent layout specifies axis suboptions such as TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

See “boolean” on page 1479 for other Boolean values that you can use.

YBINSIZE=positive-number
specifies the size of bins along the Y role, in data units. The system determines the NYBINS= and YBINSTART= values if they are not specified. The bins always span the Y data range.

Default Determined by the system.
Restriction Starting with SAS 9.4M1, the number of bins is limited to approximately 100,000. Starting with SAS 9.4M4, the limit is
specified by the ODS GRAPHICS statement option NXYBINSMAX=, which defaults to 100,000. If the total number of bins that are computed from the data, the YBINSTART= value, and the YBINSIZE= value exceeds the limit, SAS computes a new bin-size value that yields approximately the maximum number of bins. A warning of the change is written to the SAS log in that case.

Interaction  The YGAP= option is applied after this option.

Tip  Starting with SAS 9.4M4, you can use the ODS GRAPHICS statement NXYBINSMAX= option to change the limit on the number of bins. See NXYBINSMAX= on page 1494.

**YBINSTART=number**
specifies the data value for the first bin of the Y role. The system determines the NYBINS= and YBINSIZE= values if they are not specified. The bins always span the Y data range.

Default  Determined by the system.

Interaction  The YVALUES= option specifies how this value is interpreted.

**YBOUNDARY=UPPER | LOWER**
specifies how an input value is counted when it lies on the endpoint of an Y bin. If this option is set to UPPER, then the value is counted as one of the values in the upper bin. Otherwise, it is counted in the lower bin.

Default  UPPER

**YENDLABELS=TRUE | FALSE**
specifies whether the Y-axis ticks and tick values are placed at the bin endpoint or at the bin mid-points.

Default  FALSE. The axis ticks and tick values are placed at the bin mid-points.

Restriction  This option is ignored when the Y axis is discrete.

Interaction  This option is ignored when YBINAXIS=FALSE is in effect.

Tip  The axis ticks and tick value placements are independent of the YVALUES= option.

**YGAP=number**
specifies the amount of horizontal space on either side of each color-coded region in the heat map.

Default  0

Interaction  If the YBINSIZE= option is specified, the YGAP= value is subtracted from the specified Y bin size.

**YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS**
specifies whether the YBINSTART= value represents the midpoint, lower endpoint, or upper endpoint of the bin.

Default  MIDPOINTS
Example: HEATMAP Statement

The following graph was generated by the "Example Program" on page 511.

```
proc template;
  define statgraph heatmap;
  begingraph;
    entrytitle "Vehicle Mileage By Curb Weight";
    layout overlay /
      xaxisopts=(label="Curb Weight (LBS)");
    heatmap x=weight y=mpg_city / name="heatmap"
      nbins=11 ybinsize=5
      nxbins=11 xbinsize=500;
    continuouslegend "heatmap" / title="Count"
      location=outside;
  endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template=heatmap;
run;
```
HEATMAPPARM Statement

Creates a two-dimensional plot that represents the values of three variables. Generating an X, Y grid of rectangles from the values of two independent variables, it colors the rectangles to represent the values of a third variable, which can be a response variable or a group variable.

Restriction:  Starting with SAS 9.4M4, the number of bins is limited to 100,000. If the number of bins exceeds the maximum, the heat map is not drawn. A warning message is written to the SAS log in that case.

Requirements:  The COLORGROUP= or the COLORRESPONSE= role must be specified.

The data for a parameterized heat map in a single-cell graph or in a cell in a classification panel must have at least two bins for both the X and Y axes. Otherwise, the heat map is not drawn.

Note:  The data for a parameterized heat map should contain only one observation for each X and Y value pair.

Tips:  Starting with SAS 9.4M4, the limit on the number of bins is specified by the ODS GRAPHICS statement NXYBINSMAX= option. The default value is 100,000. See NXYBINSMAX= on page 1494.

Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

HEATMAPPARM X=column | expression
Y=column | expression
COLORGROUP=column | discrete-attr-var | expression <options>;

HEATMAPPARM X=column | expression
Y=column | expression
COLORRESPONSE=numeric-column | range-attr-var | expression <options>;

Summary of Optional Arguments

Appearance options

  - COLORMODEL=color-ramp-style-element | (color-list)
    specifies a color ramp to use with the COLORRESPONSE= option.

  - DATATRANSPARENCY=number
    specifies the degree of the transparency of the filled rectangles.

  - DISPLAY=(display-options) | STANDARD | ALL
    specifies which region features to display.

  - FILLATTRS=(TRANSPARENCY=number)
    specifies the transparency of the area fill in the rectangles.

  - INCLUDEMISSINGCOLOR=TRUE | FALSE
specifies whether missing values of the color-group variable or of the color-response variable are included in the plot.

OLTLINEATTRS=style-element | style-element (line-options) | (line-options)

specifies the appearance of the outlines of the filled rectangles.

REVERSECOLORMODEL=TRUE | FALSE

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

XGAP=number

specifies the amount of horizontal space on either side of each filled rectangle.

XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS

specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

YGAP=number

specifies the amount of vertical space on either side of each filled rectangle.

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS

specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Axes options

PRIMARY=TRUE | FALSE

specifies that the data columns for this plot and the plot type be used for determining default axis features.

XAXIS=X | X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

XBINAXIS=TRUE | FALSE

specifies whether to use bins as the basis for X-axis tick marks.

XBOUNDARIES=(numeric-list)

specifies the boundaries of the X-value bins.

YAXIS=Y | Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

YBINAXIS=TRUE | FALSE

specifies whether to use bins as the basis for Y-axis tick marks.

YBOUNDARIES=(numeric-list)

specifies the boundaries of the Y-value bins.

Data option

DISCRETEX=TRUE | FALSE

specifies whether the X axis is discrete when X= specifies a numeric column.

DISCRETEY=TRUE | FALSE

specifies whether the Y axis is discrete when Y= specifies a numeric column.

Data tip options

ROLENAME=(role-name-list)

specifies user-defined roles that can be used to display information in the data tips.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a filled rectangle.

**TIPFORMAT=(role-format-list)**
specifies display formats for tip columns.

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns.

**Label options**

**XENDLABELS=** TRUE | FALSE
specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

**YENDLABELS=** TRUE | FALSE
specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

**ODS options**

**URL=** string-column
specifies an HTML page to display when a rectangle is selected.

**Plot reference options**

**NAME=** "string"
assigns a name to this plot statement for reference in other template statements.

**Required Arguments**

**X=** column | expression
specifies the column for the X values.

**Requirement**
If X= specifies a numeric column, then the DISCRETEX= option must be set correctly for the X-axis type. When it is not set correctly, the heat map is not drawn. If the X-axis type is discrete, then you must specify DISCRETEX=TRUE in the HEATMAP statement. Otherwise, DISCRETEX=FALSE must be in effect.

**See** DISCRETEX= on page 516

**Y=** column | expression
specifies the column for the Y values.

**Requirement**
If Y= specifies a numeric column, then the DISCRETEY= option must be set correctly for the Y-axis type. When it is not set correctly, the heat map is not drawn. If the Y-axis type is discrete, then you must specify DISCRETEY=TRUE in the HEATMAP statement. Otherwise, DISCRETEY=FALSE must be in effect.

**See** DISCRETEY= on page 516

**COLORGROUP=** column | discrete-attr-var | expression
specifies a column or a discrete attribute variable that is used to discretely color the regions in the heat map.

**discrete-attr-var**
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

This argument is required when the response variable is of type discrete.

If you specify a column in a SAS data set, colors are mapped to group values in data order. If you specify a column in a CAS in-memory table, colors are mapped to group values in ascending order of the group column character values or unformatted numeric values.

“DISCRETEATTRVAR Statement” on page 1435

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies a numeric column or a range attribute variable that is used to color the regions of the heat map.

range-attr-var
specifies a range attribute variable that is defined in a RANGEATTRVAR statement.

A range attribute variable specification must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

“RANGEATTRVAR Statement” on page 1448

When numeric-column or expression is specified, the colors for each region are computed by mapping the values to a color ramp that is defined by the COLORMODEL= option. When attr-variable is specified, the colors defined in the associated RANGEATTRVAR or DISCRETEATTRVAR statement are used to color the regions.

This argument is required when the response variable is of type interval.

“RANGEATTRVAR Statement” on page 1448

Optional Arguments

COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

color-ramp-style-element
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

STARTCOLOR
specifies the color for the smallest data value of the COLORRESPONSE= column.

NEUTRALCOLOR
specifies the color for the midpoint of the range of the COLORRESPONSE= column.
ENDCOLOR

specifies the color for the highest data value of the COLORRESPONSE= column.

(color-list)

specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement  The list of colors must be enclosed in parentheses.

See  “color ” on page 1480

Default  The ThreeColorRamp style element

Interaction  For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip  To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

DATATRANSPARENCY=number

specifies the degree of the transparency of the filled rectangles.

Default  0

Range  0–1, where 0 is opaque and 1 is entirely transparent

DISCRETEX=TRUE | FALSE

specifies whether the X axis is discrete when X= specifies a numeric column.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default  FALSE

Requirements  If X= specifies a numeric column and the X-axis type is discrete, then you must specify DISCRETEX=TRUE. Otherwise, the heat map might not be drawn.

In SAS programs that were written before SAS 9.4M3, if the HEATMAPPARAM statement is used to plot data on a discrete category axis, you must add the DISCRETEX=TRUE option to the HEATMAPPARAM statement. Otherwise, the heat map is not drawn.

Interactions  If X= specifies a character column, then this option is ignored, and the X axis is considered to be discrete.

When this option is set to TRUE, the XENDLABELS= option is ignored.

See  X= on page 514

DISCRETEY=TRUE | FALSE

specifies whether the Y axis is discrete when Y= specifies a numeric column.
Note: This feature applies to SAS 9.4M3 and to later releases.

Default

**FALSE**

Requirements

If `Y=` specifies a numeric column and the Y-axis type is discrete, then you must specify `DISCRETEY=TRUE`. Otherwise, the heat map might not be drawn.

In SAS programs that were written before SAS 9.4M3, if the HEATMAPPARM statement is used to plot data on a discrete response axis, you must add the `DISCRETEY=TRUE` option to the HEATMAPPARM statement. Otherwise, the heat map is not drawn.

Interactions

If `Y=` specifies a character column, then this option is ignored, and the Y axis is considered to be discrete.

When this option is set to TRUE, the `YENDLABELS=` option is ignored.

See

`Y=` on page 514

**DISPLAY=**(display-options) | STANDARD | ALL

specifies which region features to display.

*display-options*

a space-separated list of options, enclosed in parentheses.

- Currently, only OUTLINE is supported, which displays outlined, filled rectangles.
  - **STANDARD**
    - displays the region fill color
  - **ALL**
    - displays the region fill color and outline

Default

**STANDARD**

**FILLATTRS=**(TRANSPARENCY=number)

specifies the transparency of the area fill in the rectangles.

Default

The **DATATRANSPARENCY**= option value

Range

0–1, where 0 is opaque and 1 is entirely transparent

Restriction

Only the TRANSPARENCY= suboption is honored. If a style element or any other fill suboption is specified, then it is ignored by the HEATMAPPARM statement.

Note

The fill colors are determined by the **COLORRESPONSE**= or **COLORGROUP**= or column.

**INCLUDEMISSINGCOLOR=**TRUE | FALSE

specifies whether missing values of the color-group variable or of the color-response variable are included in the plot.
Note: This feature applies to SAS 9.4M2 and to later releases.

Default     TRUE

Tip     The attributes of the missing color-group or color-response value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See     COLORGROUP= on page 514
        COLORRESPONSE= on page 515

“boolean ” on page 1479 for other Boolean values that you can use.

NAME="string"

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)

specifies the appearance of the outlines of the filled rectangles.

Default     The ContrastColor and LineThickness attributes of the GraphOutlines style element.

Interaction  For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

See     “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
        “Line Options” on page 1522 for available line-options.

PRIMARY=TRUE | FALSE

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default     FALSE

Restriction  This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note     In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same
axis, then the last such plot encountered is considered the primary plot.

See

“When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**REVERSECOLORMODEL=TRUE | FALSE**

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

**Default** FALSE

**See**

COLORMODEL= “boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

**(role-name-list)**

a space-separated list of role-name = column pairs.

**Example**

The following example assigns the column Obs to the user-defined role TIP:

ROLENAME=(TIP1=OBS)

**Default** No user-defined roles

**Requirement**

The role names that you choose must be unique and different from the predefined roles X, Y, COLORGROUP=, and COLORRESPONSE=.

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over a filled rectangle. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the heat map can be specified along with roles that do.

**(role-list)**

an ordered, space-separated list of unique HEATMAPPARM and user-defined roles. HEATMAPPARM roles include X and Y.

**Example**

The following example displays data tips for the columns X and PCT. The Pct column is not assigned to any pre-defined HEATMAPPARM role, so it must first be assigned a role:

ROLENAME=(TIP1=PCT)

TIP=(X TIP1)

**NONE**

suppresses data tips from the plot.

**Default**

The columns assigned to these roles are automatically included in the data tip information: X and Y, and COLORGROUP= or COLORRESPONSE=.
Requirement  To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction  This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip  The labels and formats for the TIP variables can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)  specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)  a space-separated list of role-name = format pairs.

Example  ROLENAME=(TIP1=SALARY)
          TIP=(TIP1)
          TIPFORMAT=(TIP1=DOLLAR12.)

Default  The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction  Only the roles that appear in the TIP= option are used.

Requirement  A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TIPLABEL=(role-label-list)  specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list  a space-separated list of rolename = "string" pairs.

Example  ROLENAME=(TIP1=PCT)
          TIP=(TIP1)
          TIPLABEL=(TIP1="Percent")

Default  The column label or column name of the column assigned to the role.

Restriction  Only the roles that appear in the TIP= option are used.

Requirement  A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

URL=string-column  specifies an HTML page to display when a rectangle is selected.

string-column  specifies a column that contains a valid HTML page reference (HREF) for each rectangle that is to have an active link.

Example  http://www.sas.com
To generate a plot with selectable rectangles, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

The URL value can be blank for some X and Y pairs. In that case, no action is taken when the corresponding rectangle is selected.

The URL value can be the same for any X and Y pairs. In that case, the same action is taken when the rectangle for those X and Y pairs is selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Interaction The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

XBINAXIS=TRUE | FALSE
specifies whether to use bins as the basis for X-axis tick marks. When this option is FALSE, a standard axis is used, ignoring bin boundaries and midpoints.

Default TRUE

Interaction When this option is TRUE, the XENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is FALSE, the XENDLABELS= option is ignored.

When this option is TRUE, the axis ticks are in predetermined locations and are not changed when the parent layout specifies axis suboptions such as TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

See "boolean " on page 1479 for other Boolean values that you can use.

XBOUNDARIES=(numeric-list)
specifies the boundaries of the X-value bins. The boundaries are specified as a space-separated list of values enclosed in parentheses. The keywords MIN and MAX can be used as one of the values in the list of boundaries. Keywords MIN and MAX indicate the minimum and maximum data values for the X variable.
Interaction  This option is ignored if the X values are not numeric.

Tip  This option can be used to specify unequal bins.

Example  xboundaries=(MIN 20 200 250 MAX)

**XENDLABELS=TRUE | FALSE**
specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

**Default**  FALSE. The axis ticks and values labels are drawn at the bin midpoints.

**Restriction**  This option is ignored when the X axis is discrete.

**Interactions**  If this option is set to FALSE, then the axis ticks and value labels are drawn at the bin midpoints, regardless of whether the XVALUES= option identifies the X data as endpoint values or midpoint values.

This option is ignored when XBINAXIS=FALSE is in effect.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**XGAP=number**
specifies the amount of horizontal space on either side of each filled rectangle.

**Default**  0

**XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS**
specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

**Default**  MIDPOINTS

See  XENDLABELS=

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

**Default**  Y

**Interaction**  The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YBINAXIS=TRUE | FALSE**
specifies whether to use bins as the basis for Y-axis tick marks. When this option is FALSE, a standard axis is used, ignoring bin boundaries and midpoints.

**Default**  TRUE

**Interactions**  When this option is TRUE, the YENDLABELS= option determines how the axis ticks and value labels are displayed. When this option is FALSE, the YENDLABELS= option is ignored.

When this option is TRUE, the axis ticks are in predetermined locations and are not changed when the parent layout specifies axis
suboptions such as TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

See "boolean " on page 1479 for other Boolean values that you can use.

YBOUNDARIES=(numeric-list)
specifies the boundaries of the Y-value bins. The boundaries are specified as a space-separated list of values enclosed in parentheses. The keywords MIN and MAX can be used as one of the values in the list of boundaries. Keywords MIN and MAX indicate the minimum and maximum data values for the Y variable. Example:

```
yboundaries=(MIN 20 200 250 MAX)
```

Interaction This option is ignored if the Y values are not numeric.

Tip This option can be used to specify unequal bins.

YENDLABELS=TRUE | FALSE
specifies whether the axis ticks and value labels are drawn at the endpoints of the bins or midpoints of the bins.

Default FALSE. The axis ticks and values labels are drawn at the bin midpoints.

Restriction This option is ignored when the Y axis is discrete.

Interactions If this option is set to FALSE, then the axis ticks and value labels are drawn at the bin midpoints, regardless of whether the YVALUES= option identifies the Y data as endpoint values or midpoint values.

This option is ignored when YBINAXIS=FALSE is in effect.

See "boolean " on page 1479 for other Boolean values that you can use.

YGAP=number
specifies the amount of vertical space on either side of each filled rectangle.

Default 0

YVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
specifies whether the input Y values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default MIDPOINTS

See YENDLABELS=

Details
A heat map is useful for visualizing the magnitude of a response variable in relation to two independent variables. For example, in molecular biology, heat maps can be used to track the expression of genes across multiple sample studies. In the HEATMAPPARM statement, you specify the independent variables in the X and Y arguments. For a response variable that has discrete values, use the COLORGROUP argument, or for a response variable that has interval values, use
the COLORRESPONSE argument. For interval response variables, you can use the COLORMODEL= option to define the color ramp that is used to fill the rectangles.

Example: HEATMAPPARM Statement

The following graph was generated by the “Example Program” on page 524:

Example Program

```
proc template;
    define statgraph heatmapparm;
    begingraph;
    layout overlay;
        heatmapparm x=height y=weight colorresponse=count /
            name="heatmapparm" xbinaxis=false ybinaxis=false;
        // Customize the legend
        continuouslegend "heatmapparm" / location=outside valign=bottom;
    endlayout;
    endgraph;
end;
run;

proc sgrender data=sashelp.gridded template=heatmapparm;
run;
```
HIGHLOWPLOT Statement

Creates a display of floating vertical or horizontal lines or bars that connect the minimum and maximum response values for each value of a categorical variable.

Requirements: Either the X= or Y= argument must be specified, but you cannot specify both on the same HIGHLOWPLOT statement.

The HIGH= and LOW= arguments are required.

Note: Specifying the X= option creates a vertical high-low chart, which would typically be used in the financial industry to plot stock values over time. Specifying the Y= option creates a horizontal high-low chart, which would typically be used in the Health and Life Sciences industry to display over time the duration of adverse events or of adverse reactions to medication.

Tips: For charts that have a large number of bars that are very close together, slight variations in spacing that normally occur due to integer rounding can become more obvious. Subpixel rendering provides more precise bar spacing in that case. In SAS 9.4M2 and in earlier releases, specify SUBPIXEL=ON in the BEGINGRAPH statement to enable subpixel rendering. See SUBPIXEL= on page 35. Starting with SAS 9.4M3, subpixel rendering is enabled by default.

To disable subpixel rendering in SAS 9.4M3 and in later releases, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

HIGHLOWPLOT X=column | expression
LOW=numeric-column | expression
HIGH=numeric-column | expression </options>;

HIGHLOWPLOT Y=column | expression
LOW=numeric-column | expression
HIGH=numeric-column | expression </options>;

Summary of Optional Arguments

Appearance options

BARWIDTH=number
    specifies the width of a bar as a ratio of the maximum possible width.

CLIPCAP=TRUE | FALSE
    specifies whether a special clip cap is displayed to indicate where clipping occurred.

CLIPCAPSHAPE=DEFAULT | BARBEDARROW | CLIPPEDARROW | CLOSEDARROW | FILLEDARROW | OPENARROW | SERIF
    specifies the shape of the arrowhead on the clipped end of a line or bar when CLIPCAP=TRUE.

CLOSE=numeric-column | expression
specifies a column or expressions whose values are used to display a closing-value indicator.

**CLUSTERWIDTH=number**

on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

**COLORMODEL=color-ramp-style-element | (color-list)**

specifies a color ramp to use with the COLORRESPONSE= option.

**COLORRESPONSE=numeric-column | range-attr-var | expression**

specifies the column or range attribute variable to use to map the bar or line colors.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**

enhances the visual appearance of the filled bars or lines of a high-low chart.

**DATATRANSPARENCY=number**

specifies the degree of the transparency of the colored regions, and the high-end and low-end labels.

**DISPLAY=(display-options) | STANDARD | ALL**

specifies which bar features to display.

**ENDCAPDISPLAYPOLICY=AUTO | ALWAYS**

specifies the policy for displaying end caps when end caps are present.

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**

specifies the appearance of the interior fill area of the bars.

**FILLPATTERNATTRS=style-element | (fill-pattern-options)**

specifies the appearance of the pattern-filled areas.

**HIGHCAP=column | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW**

specifies the type of cap used at the high end of the bar or line.

**INDEX=positive-integer-column | expression**

specifies indices for mapping line attributes (color and line pattern) or bar attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**INTERVALBARWIDTH=dimension**

specifies the width of the floating bars.

**LINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the floating plot lines.

**LOWCAP=column | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW**

specifies the type of cap used at the low end of the bar or line.

**OPEN=numeric-column | expression**

specifies a column or expressions whose values are used to display an opening-value indicator.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the appearance of the outlines of the filled regions.

**TYPE=LINE | BAR**

specifies whether data values should be represented by bars or lines.

**Axes options**

**PRIMARY=TRUE | FALSE**

specifies that the data columns for this plot and the plot type be used for determining default axis features.
XAXIS=X | X2
   specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
   specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options
   ROLENAME=(role-name-list)
      specifies user-defined roles that can be used to display information in the data tips.
   TIP=(role-list) | NONE
      specifies the information to display when the cursor is positioned over a bar or line.
   TIPFORMAT=(role-format-list)
      specifies display formats for tip columns.
   TIPLABEL=(role-label-list)
      specifies display labels for tip columns.

Label options
   HIGHLABEL=column | expression
      specifies the label to display at the high end of the bar or line.
   LABELATTRS=style-element | style-element (text-options) | (text-options)
      specifies the color and font attributes of the labels for the bars or lines.
   LEGENDLABEL="string"
      specifies a label to be used in a discrete legend for this plot.
   LOWLABEL=column | expression
      specifies the label to appear at the low end of each floating bar or line.

Midpoint options
   DISCRETEOFFSET=number
      specifies an amount to offset all bars or lines from the category midpoints when graphing multiple response variables side by side on a common axis.
   GROUP=column | discrete-attr-var | expression
      creates a distinct set of floating bars or lines for each unique group value in the specified column.
   GROUPDISPLAY=OVERLAY | CLUSTER
      specifies whether grouped bars or lines are overlaid or clustered around the category midpoints.
   GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
      specifies the ordering of the groups within a category.
   INCLUDEMISSINGGROUP=TRUE | FALSE
      specifies whether missing values of the group variable are included in the plot.

Plot reference options
   NAME="string"
      assigns a name to this plot statement for reference in other template statements.
Required Arguments

Either the X= or Y= argument must be specified, but you cannot specify both on the same HIGHLOWPLOT statement. The HIGH= and LOW= arguments are required.

\[X=\text{column }|\text{ expression}\]
specifies a column or expression representing the values in a vertical high-low chart. The values can be character or numeric.

\[Y=\text{column }|\text{ expression}\]
specifies a column or expression representing the values in a horizontal high-low chart. The values can be character or numeric.

\[\text{LOW=}\text{numeric-column }|\text{ expression}\]
specifies a column or expression representing the values of the lower end of the floating bar or line.

\[\text{HIGH=}\text{numeric-column }|\text{ expression}\]
specifies a column or expression representing the values of the higher end of the floating bar or line.

Optional Arguments

\[\text{BARWIDTH=}\text{number}\]
specifies the width of a bar as a ratio of the maximum possible width.

| Default | 0.85 |
| Range   | 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width |
| Restriction | This option has an effect only when TYPE= BAR. |
| Interaction | Prior to SAS 9.4M3, this option is ignored for an interval high-low plot, and the bar width is controlled by the INTERVALBARWIDTH= option. Starting with SAS 9.4M3, this option is honored for an interval high-low plot but it can be overridden by the INTERVALBARWIDTH= option. |
| Notes | This option is needed only to change the default behavior. |
| Tip | To remove any inter-bar gap, set BARWIDTH=1. |
| See | DISCRETEOFFSET= option for examples of using this option. |

\[\text{CLIPCAP=}\text{TRUE }|\text{ FALSE}\]
specifies whether a special clip cap is displayed to indicate where clipping occurred. When the VIEWMIN= and VIEWMAX= axis options are specified for an axis and a data value exceeds the specified axis range, the bar for that value is clipped. In that case, when CLIPCAP=TRUE, a special clip cap is displayed at the clipped end of the bar. If the bar already has a high or low cap, then it is replaced by the clip cap.

For vertical bars, the clip cap is added to the end of the bar that is clipped by the Y-axis range. The default clip cap is a vertical clipped arrowhead that points toward the clip edge (🍃 or 🔧). For horizontal bars, the cap is added to the end of the bar that is clipped by the X-axis range. The default clip cap is a horizontal
clipped arrowhead that points toward the clip edge (« or »). If an entire bar is clipped, then a clip cap is displayed at the high or low side where the bar was clipped.

The following figure shows a side-by-side comparison of a vertical line high-low chart with no clipping and with clipping when CLIPCAP=TRUE.

The first graph shows the default case in which no clipping occurs. The second graph shows the case in which VIEWMIN= and VIEWMAX= specify a range that causes clipping on the Y axis. In the second graph, the clip caps indicate clipping for each category line as follows:

A the high end of the line is clipped.
B the low end of the line is clipped.
C both ends of the line are clipped.
D the entire line is clipped at the low end.
E the entire line is clipped at the high end.

Default FALSE

Interaction Clip caps appear only when CLIPCAP=TRUE and the data values exceed the axis range that is specified by the VIEWMIN= and VIEWMAX= options. When the VIEWMIN= and VIEWMAX= options are not specified, the axis range is adjusted to accommodate the data values and clipping does not occur.

Note If the HIGHLABEL= and LOWLABEL= options are in effect and the bar is clipped, then the label value at the clipped end is drawn at the tip of the clip cap. If the entire bar is clipped, then the labels are not shown.

Tip Use the CLIPCAPSHAPE= option to specify a different clip-cap arrowhead.

See “boolean ” on page 1479 for other Boolean values that you can use.
CLIPCAPSHAPE=DEFAULT | BARBEDARROW | CLIPPEDARROW | CLOSEDARROW | FILLEDARROW | OPENARROW | SERIF

specifies the shape of the arrowhead on the clipped end of a line or bar when CLIPCAP=TRUE. The following figure shows each of the clip-cap arrowhead shapes for vertical lines and bars that are clipped at both ends.

![Arrowhead Shapes](image)

In SAS 9.4M1 and earlier releases, this option specifies the arrowhead shape only for the clipped end of clipped lines (TYPE=LINE). For bars (TYPE=BAR), this option is ignored, and CLIPPEDARROW is always used as the arrowhead shape for the clipped end of clipped bars.

Starting with SAS 9.4M2, this option specifies the arrowhead shape for the clipped end of clipped lines or clipped bars. For bars, you can specify the CLIPPEDARROW shape for filled and unfilled bars. When any value other than CLIPPEDARROW or DEFAULT is specified for bars, the FILLEDARROW shape is used for filled bars, and the CLOSEDARROW shape is used for unfilled bars.

Default

DEFAULT, which is the same as CLIPPEDARROW

Interaction

This option is ignored when CLIPCAP=FALSE.

CLOSE=numeric-column | expression

specifies a column or expressions whose values are used to display a closing-value indicator. This option is typically used when TYPE=LINE, but it can be used when TYPE=BAR. For vertical high-low charts, the value is represented by a short horizontal line extending from the side that displays the higher X values. For horizontal high-low charts, the value is represented by a short vertical line extending from the side that displays the higher Y values.

CLUSTERWIDTH=number

on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

Default

0.85

Range

0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Interaction

For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.
COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

color-ramp-style-element
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

STARTCOLOR
specifies the color for the smallest data value of the COLORRESPONSE= column.

NEUTRALCOLOR
specifies the color for the midpoint of the range of the COLORRESPONSE= column.

ENDCOLOR
specifies the color for the highest data value of the COLORRESPONSE= column.

(color-list)
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement  The list of colors must be enclosed in parentheses.

See  “color ” on page 1480

Defaults  For lines or for outline-only bars, the ThreeColorAltRamp style element.

For bars with fill, the ThreeColorRamp style element

Interaction  For this option to take effect, the COLORRESPONSE= option must also be specified.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the column or range attribute variable to use to map the bar or line colors.

Note:  This feature applies to SAS 9.4M3 and to later releases.

range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“RANGEATTRVAR Statement”

Restriction  A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See  “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.
Requirement
For a grouped plot, the COLORRESPONSE values should remain constant for each group value. If the COLORRESPONSE column has multiple values for a single GROUP value, unexpected results might occur.

Interactions
When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

When fill is displayed, this option overrides suboption COLOR= in the FILLATTRS= option and varies the fill color according to the color gradient or the attribute map.

When only the outlines are displayed, this option overrides suboption COLOR= in the OUTLINEATTRS= option and varies the outline color according to the color gradient or the attribute map.

Tips
To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

For a numeric column or expression, the ThreeColorRamp style element defines the fill color gradient, and the ThreeColorAltRamp style element defines the outline color gradient.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**

enhances the visual appearance of the filled bars or lines of a high-low chart. The following figure shows a high-low chart that contains a filled bar and a line with each of the skins applied.

![Diagram showing the effects of different skin styles](image)

Default
The **DATASKIN=** option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

Restriction
Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the **DATASKINMAX=** option in your ODS GRAPHICS statement to increase the maximum limit.
Interactions

This option overrides the BEINGRAPH statement DATASKIN= option.

When TYPE=BAR, the DISPLAY= option in effect must include FILL in order for the DATASKIN= option to have any effect.

For filled bars, the skin appearance is based on the FILLATRWS= option color specification.

**DATATRANSPARENCY=number**

specifies the degree of the transparency of the colored regions, and the high-end and low-end labels.

- **Default** 0
- **Range** 0–1, where 0 is opaque and 1 is entirely transparent

**Tip**
The FILLATRWS option can be used to set transparency for just the colored regions. You can combine this option with FILLATRWS= to set one transparency for the outlines but a different transparency for the region fills. Example:
```
DATATRANSPARENCY=0.2 FILLATRWS=(TRANSparency=0.6)
```

**DISCRETEOFFSET=number**

specifies an amount to offset all bars or lines from the category midpoints when graphing multiple response variables side by side on a common axis.

- **Default** 0 (no offset, all bars or lines are centered on the category midpoints)
- **Range** –0.5 to +0.5, where 0.5 represents half the distance between category ticks. Normally, a positive offset is to the right for a vertical plot and up for a horizontal plot. (If the layout's axis options set REVERSE=TRUE, then the offset direction is also reversed.)

**Restriction**
This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

**Tip**
Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**See**
“About the DISCRETEOFFSET= Option” on page 547

**DISPLAY=(display-options) | STANDARD | ALL**

specifies which bar features to display.

- **(display-options)** specifies a space-separated list of one or more display options, enclosed in parentheses.

  The list can contain one or more of the following options:

  **FILL**
  
  displays the bar fill color
OUTLINE
displays the bar outline

FILLPATTERN
displays the bar fill pattern

Note This feature applies to SAS 9.4M5 and to later releases.

Tip By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the FILLPATTERNAATTRS= option to specify a fill pattern for the bars.

STANDARD
specifies FILL and OUTLINE

ALL
specifies all features: FILL, FILLPATTERN, and OUTLINE

Defaults In SAS 9.4M4 and in earlier releases, STANDARD

Requirement You must also specify TYPE=BAR. Otherwise, this option is ignored.

Tip Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the bar outline and fill color. Starting with SAS 9.4M5, use the FILLPATTERNAATTRS= option to control the appearance of the bar fill pattern.

ENDCAPDISPLAYPOLICY= AUTO | ALWAYS
specifies the policy for displaying end caps when end caps are present.

AUTO
draws the end caps only for the elements with a low-to-high range that is large enough to accommodate the end caps.

Draws just the bar or line for the rest.

ALWAYS
always draws the end caps.

Note When the low-to-high range is not large enough for the end caps, the end caps might overlap. The data ranges that are smaller than the end-cap size might not be resolvable beyond the rendered size of the end caps.

Default AUTO

Interactions This option is honored only when the option TYPE=LINE is in effect.

This option is ignored if the LOWCAP= or HIGHCAP= option is not set or is effectively set to NONE.

FILLATTRS= style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the interior fill area of the bars.
### FILLPATTERNATTRS

**Defaults**
For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

**Interactions**
For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

Starting with **SAS 9.4M5**, when COLORRESPONSE= is in effect and the DISPLAY= option enables FILLPATTERN display, the FILLPATTERNATTRS= suboption COLOR= is ignored, and the fill-pattern colors vary according to the gradient.

**Tip**
In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITIZE=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in **SAS Graph Template Language: User’s Guide**.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Pattern Options” on page 1521 for available fill-pattern-options.
GROUP=column | discrete-attr-var | expression
creates a distinct set of floating bars or lines for each unique group value in the
specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR
statement.

Restriction  A discrete attribute map variable specification must be a direct
reference to the attribute map variable. It cannot be set by a
dynamic variable.

When TYPE= on page 545LINE (default), combinations of line color and line
style are used to create a distinct line for each group value. The ContrastColor
and LineStyle attributes of a GraphData1–GraphDataN style element or the
GraphMissing style element control the color and line style of each group line.

When TYPE=BAR, a distinct bar is created for each group value by varying the
visual attributes of the bar display features. The display features are controlled
by the current ODS style or by the DISPLAY= option. The default group
appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature 1</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern 2</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern.</td>
</tr>
</tbody>
</table>
| Outline           | - if the outline is enabled with fill color, fill pattern, or both, ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style  
                    - if the outline is the only display feature enabled, ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element |

1 Fill color and the bar outline are displayed by default.
2 Fill patterns are valid for high-low plots starting with SAS 9.4M5.

Interactions  If the X or Y role is discrete, then the bars or lines representing the
group can be drawn in a cluster or overlaid, based on the setting for the GROUPDISPLAY= option. However, if the X or Y role is interval, then the lines representing the group are overlaid and the GROUPDISPLAY= option is ignored.
By default, the group values are mapped in the order of the data. Use the GROUPORDER= option to control the sorting order of the grouped bar segments. Use the INDEX= option to alter the default sequence of colors and line patterns.

The INCLUDEMISSINGGROUP= option determines whether missing group values are considered a distinct group value.

When both the GROUP= and COLORRESPONSE= options are specified, the color attributes are controlled by the COLORRESPONSE= option.

**Note**
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tip**
The representations that are used to identify the groups can be overridden. For example, if each distinct group value is represented by a different line pattern, you can use the LINEATTRS=(PATTERN=pattern) specification to assign the same line pattern to all the plot lines. In that case, the line color denotes group values. Likewise, you can use LINEATTRS=(COLOR=color) to assign the same color to all lines, letting line pattern denote group values.

See “DISCRETEATTRVAR Statement” on page 1435

**GROUPDISPLAY=OVERLAY | CLUSTER**
specifies whether grouped bars or lines are overlaid or clustered around the category midpoints.

**OVERLAY**
centers the bars or lines for matching category values on the midpoints.

The bars or lines in each set of group values are superimposed on each other.

**CLUSTER**
clusters the bars or lines for matching category values around the midpoints.

Each cluster of group values is centered at the midpoint for the category.

The following example shows the effect of clustering the lines in a stock report when the category values are grouped into a single response variable. Note that if your category values are not grouped in the same column but are stored in separate columns, then you can get this same effect by using the DISCRETEOFFSET= option.

```sas
layout overlay /
    yaxisopts=(label="Stock Value");
highlowplot x=month high=high low=low /
    close=close open=open
    legendlabel="Stock" name="cluster"
    group=stock groupdisplay=cluster
    lineattrs=(pattern=solid);
    discretelegend "cluster" / title="Stock"
    location=inside halign=right valign=top;
endlayout;
```
For this option to take effect, the GROUP= option must also be specified.

When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

**GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING**

specifies the ordering of the groups within a category.

**DATA**
orders the groups within a category in the group-column data order.

**REVERSEDATA**
orders the groups within a category in the reverse group-column data order.

**ASCENDING**
orders the groups within a category in ascending order.

**DESCENDING**
orders the groups within a category in descending order.

**Defaults**
DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

**Interactions**
This option is ignored if the GROUP= option is not also specified.

This option is ignored when GROUPDISPLAY=OVERLAY.
By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

\texttt{HIGHCAP=column} | \texttt{NONE} | \texttt{SERIF} | \texttt{BARBEDARROW} | \texttt{FILLEDARROW} | \texttt{OPENARROW} | \texttt{CLOSEDARROW}

specifies the type of cap used at the high end of the bar or line. All of the keywords can be specified for any high-low chart. The effect of each keyword depends on the setting for the TYPE= and DISPLAY= options as follows:

- When TYPE=BAR and DISPLAY= includes FILL, FILLEDARROW is used for all settings other than NONE.
- When TYPE=BAR and DISPLAY= does not include FILL, CLOSEDARROW is used for all settings other than NONE.
- When TYPE=LINE and CLOSEDARROW is specified, FILLEDARROW is used instead.

The following figure shows the effect of each cap value on horizontal lines, filled bars, and unfilled bars.
Horizontal High and Low Cap Shapes for Lines, Filled Bars, and Unfilled Bars

**Figure 6.1**

![Diagram of cap shapes](image)

<table>
<thead>
<tr>
<th>NOIE</th>
<th>SERIF</th>
<th>BARBEDARROW</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="No Cap" /></td>
<td><img src="image" alt="Serif Cap" /></td>
<td><img src="image" alt="Barbed Arrow Cap" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FILLEDARROW</th>
<th>OPENARROW</th>
<th>CLOSEDARROW</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Filled Arrow" /></td>
<td><img src="image" alt="Open Arrow" /></td>
<td><img src="image" alt="Closed Arrow" /></td>
</tr>
</tbody>
</table>

**Default**

NONE

**Interaction**

When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself is reduced.

**Note**

If the length of the high-low element is smaller than the cap, then the cap is not drawn.

**HIGHLABEL=**

- `column` | `expression`

specifies the label to display at the high end of the bar or line.

**INCLUDEMISSINGGROUP=**

- `TRUE` | `FALSE`

specifies whether missing values of the group variable are included in the plot.

**Default**

TRUE

**Interaction**

For this option to take effect, the GROUP= option must also be specified.

**Tip**

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**

“boolean ” on page 1479 for other Boolean values that you can use.

**INDEX=**

- `positive-integer-column` | `expression`

specifies indices for mapping line attributes (color and line pattern) or bar attributes (fill and outline) to one of the GraphData1–GraphDataN style elements.

**Requirements**

The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.
The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

Interaction

For this option to take effect, the GROUP= option must also be specified.

Notes

The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped according to the setting of the GROUPORDER= option.

Tip

You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**INTERVALBARWIDTH=** *dimension*

specifies the width of the floating bars.

Defaults

Prior to SAS 9.4M3, 85% of the smallest interval between any two bars for the given plot

Starting with SAS 9.4M3, the BARWIDTH= option setting

Restriction

For this option to take effect, TYPE=BAR must be set, and the independent variable must be of type interval.

Interaction

Prior to SAS 9.4M3, this option controls the bar width for a high-low plot. Starting with SAS 9.4M3, this option overrides the BARWIDTH= option.

Tip

The bar width that you specify with this option is honored even if the bars overlap. If the bars overlap, reduce the interval bar width or use the BARWIDTH= option instead.

See

“dimension” on page 1480

**LABELATTRS=** *style-element* | *style-element* *(text-options)* | *(text-options)*

specifies the color and font attributes of the labels for the bars or lines.

Defaults

In SAS 9.4M2 and earlier releases, the GraphDataText style element.

Starting with SAS 9.4M2, the GraphDataText style element is the default for non-grouped data. For grouped data, the data label color is determined by the ContrastColor attribute of the GraphData1–GraphDataN style elements.

See

“General Syntax for Attribute Options” on page 1519 for the syntax for using a *style-element*

“Text Options” on page 1525 for available *text-options*.
**LEGENDLABEL=**"string"

specifies a label to be used in a discrete legend for this plot.

**Default**
The string specified on the NAME= option.

**Restriction**
This option applies only to an associated DISCRETELEGEND statement.

**Interaction**
If the GROUP= option is specified, then this option is ignored.

**LINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the floating plot lines.

**Defaults**
For non-grouped data, the GraphDataDefault style element.

For grouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphData1–GraphDataN style elements.

**Interaction**
For this option to have any effect, TYPE= LINE must also be specified.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**LOWCAP=column | NONE | SERIF | BARBEDARROW | FILLEDARROW | OPENARROW | CLOSEDARROW**

specifies the type of cap used at the low end of the bar or line. All of the keywords can be specified for any high-low chart. The effect of each keyword depends on the setting for the TYPE= and DISPLAY= options as follows:

- When TYPE=BAR and DISPLAY= includes FILL, FILLEDARROW is used for all settings other than NONE.
- When TYPE=BAR and DISPLAY= does not include FILL, CLOSEDARROW is used for all settings other than NONE.
- When TYPE=LINE and CLOSEDARROW is specified, FILLEDARROW is used instead.

*Figure 6.4 on page 540* shows the effect of each cap value on horizontal lines, filled bars, and unfilled bars.

**Default**
NONE

**Interaction**
When TYPE=BAR, the caps are drawn to fit within the bar width. The width of the bar itself is reduced.

**Note**
If the length of the high-low element is smaller than the cap, then the cap is not drawn.

**LOWLABEL=column | expression**

specifies the label to appear at the low end of each floating bar or line.

**NAME=**"string"

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.
Restrictions  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**OPEN=numeric-column | expression**
specifies a column or expressions whose values are used to display an opening-value indicator. This option is typically used when TYPE=LINE, but it can be used when TYPE=BAR. For vertical high-low charts, the value is represented by a short horizontal line extending from the side that displays the lower X values. For horizontal high-low charts, the value is represented by a short vertical line extending from the side that displays the lower Y values.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**
specifies the appearance of the outlines of the filled regions.

**Defaults**
For non-grouped data, the ContrastColor, LineThickness, and LineStyle attributes of the GraphOutlines style element.

For grouped data when FILL is displayed, the LineThickness and LineStyle attributes of the GraphOutlines style element, and the ContrastColor attribute of the GraphData1–GraphDataN style elements.

For grouped data when FILL is not displayed, the LineThickness attribute of the GraphOutlines style element, and the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements.

**Interactions**
For this option to have any effect, TYPE=BAR must be specified, and outlines must be enabled by the ODS style or the DISPLAY= option.

When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the bar outline colors vary according to the gradient.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default** FALSE

**Restriction** This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note** In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same
axis, then the last such plot encountered is considered the primary plot.

See  

“When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

(role-name-list)  
a space-separated list of role-name = column pairs.

**Example**
The following example assigns the column Obs to the user-defined role TIP:

```
ROLENAME=(TIP1=OBS)
```

**Default**
No user-defined roles

**Requirement**
The role names that you choose must be unique and different from the predefined roles X, Y, LOW, HIGH, OPEN, and CLOSE.

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over a bar or line. If this option is used, then the information specified replaces all of the information that is displayed by default. You can specify roles for columns that do not contribute to the bar chart along with roles that do.

(role-list)  
an ordered, space-separated list of unique HIGHLOWPLOT and user-defined roles.

HIGHLOWPLOT roles include X, Y, LOW, HIGH, OPEN, CLOSE, GROUP, and COLORRESPONSE.

**Note**
The COLORRESPONSE role is valid starting with SAS 9.4M3.

**Example**
The following example displays data tips for the columns X and Pct. The Pct column is not assigned to any pre-defined HIGHLOWPLOT role, so it must first be assigned a role.

```
ROLENAME=(TIP1=PCT)
TIP=(X TIP1)
```

**NONE**
suppresses data tips from the plot.

**Default**
The columns assigned to these roles are automatically included in the data tip information: X or Y and GROUP.

**Requirement**
To generate data tips in the output, you must include an ODS GRAPHICS ON statement with the IMAGEMAP option specified, and you must write the output to the ODS HTML destination.

**Interaction**
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is
Tip You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction Only the roles that appear in the TIP= option are used.

Requirement A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

Default The column label or column name of the column assigned to the role.

Restriction Only the roles that appear in the TIP= option are used.

Requirement A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TYPE=LINE | BAR
specifies whether data values should be represented by bars or lines.

BAR
uses fill and outline attributes.

LINE
uses line attributes.

Default LINE

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
When the Y= argument is specified, the HIGH= and LOW= arguments are mapped to the axis that is specified on this option (X or X2).

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default Y

When the X= argument is specified, the HIGH= and LOW= arguments are mapped to the axis that is specified on this option (Y or Y2).

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details
Statement Description
A high-low chart specifies that floating vertical or horizontal lines or bars connect the minimum and maximum response values for each value of a categorical variable. The data should have at least two response values for every category value. Otherwise, the single value is displayed without the connecting line or bar. In the statement syntax, the X and Y arguments can specify a column with character or numeric values. The LOW and HIGH arguments must specify a numeric column.

The HIGHLOWPLOT statement can be used to create a plot showing the high and low response values for observations along a time axis. The independent variable can be interval or discrete. The HIGHLOWPLOT has two common uses:

- One typical use of a high-low chart is in the financial industry to plot stock values by day. The OPEN= and CLOSE= options are typically used in the plot, and the TYPE= option is set to LINE (the default), as illustrated in “Example 1: Vertical High-low Chart” on page 548.

- Another typical use of a high-low chart is in the Health and Life Sciences industry to display over time the duration of adverse events or of adverse reactions to medication. The HIGHCAP= and LOWCAP= options are typically used in the plot, and the TYPE= option is set to BAR, as illustrated in “Example 2: Horizontal High-low Chart” on page 548.

For both vertical and horizontal high-low charts, the data values can be grouped using the GROUP= option. For grouped data, the data values are not summarized. Each observation is plotted independently, and the line or bar segment for each category value can use different display characteristics.
About the DISCRETEOFFSET= Option

This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if multiple HIGHLOWPLOT statements are used with different response variables, then the bars or lines for matching category values are centered on the midpoints and superimposed on each other. In those cases, you can make it easier to distinguish among superimposed bars or lines by assigning a different width for them in each HIGHLOWPLOT statement in the overlay.

If you prefer to avoid superimposed bars or lines, then you can assign a different offset to each HIGHLOWPLOT statement. If desired, you can adjust the width of the bar or line in conjunction with DISCRETEOFFSET= to create narrower bars that require less width within the plot area.

The following example defines offsets for the lines in a stock report:

```plaintext
layout overlay / cycleattrs=true
  yaxisopts=(label="Stock Value");
highlowplot x=month high=a_high low=a_low / close=a_close open=a_open
  legendlabel="A" name="ahighlow" lineattrs=(pattern=solid)
  discreteoffset=-0.2 ;
highlowplot x=month high=b_high low=b_low / close=b_close open=b_open
  legendlabel="B" name="bhighlow" lineattrs=(pattern=solid)
  discreteoffset=0 ;
highlowplot x=month high=c_high low=c_low / close=c_close open=c_open
  legendlabel="C" name="chighlow" lineattrs=(pattern=solid)
  discreteoffset=+0.2 ;
discretelegend "ahighlow" "bhighlow" "chighlow" / title="Stock"
  location=inside halign=right valign=top;
edlayout;
```

![Midpoint Offsets, Offset=0.2](image)

- Stock
  - A
  - B
  - C

Month: Jul, Aug, Sep, Oct, Nov, Dec

Stock Value: $22, $24, $26, $28, $30
Examples:

Example 1: Vertical High-low Chart
The following vertical high-low chart was generated by "Example Program" on page 548:

Example Program

```
proc template;
    define statgraph highlow;
        begingraph;
            layout overlay /
                yaxisopts=(griddisplay=on label="Stock Value");
            highlowplot x=date high=high low=low /
                open=open close=close;
        endlayout;
    endgraph;
end;
```

```
proc sgrender data=sashelp.stocks template=highlow;
    where date >= '01JAN03'd and stock="IBM";
run;
```

Example 2: Horizontal High-low Chart
The following horizontal high-low chart was generated by "Example Program" on page 549:
Example Program

data highlowbar;
  length cap $ 12;
  input drug $ 1-10 low high cap $;
  datalines;
  Drug A    10   20  NONE
  Drug A    30   60  FILLEDARROW
  Drug B    20   35  NONE
  Drug B    50   75  FILLEDARROW
  Drug C    30   90  FILLEDARROW
;
proc template;
define statgraph highlowbar;
begingraph;
  entrytitle 'Medications Plot';
  layout overlay /
  xaxisopts=
    griddisplay=on
    linearopts=(viewmin=0 viewmax=100)
    display=(line ticks tickvalues)
  yaxisopts=
    griddisplay=on
    display=(line ticks tickvalues));
  highlowplot y=drug high=high low=low /
  group=drug outlineattrs=(pattern=solid)
  datatransparency=0.4
  type=bar barwidth=0.4
  highcap=cap lowcap=cap;
endlayout;
endgraph;
HISTOGRAM Statement

Creates a univariate histogram computed from input data.

**Requirement:** If you include a histogram and a density plot in your graph and both plots specify the same data column, you must either specify SCALE=DENSITY in the HISTOGRAM statement or place the HISTOGRAM statement before the DENSITYPLOT statement in your template code. Otherwise, the density plot values might not scale properly to the histogram bins on the dependent axis. This requirement applies even when the histogram and density plots are in different cells in a multicell graph.

**Note:** When a histogram is placed on a log axis, binning is done on a linear scale, but the bins are drawn on the log scale. As a result, the bins might have different widths along the log axis.

**Syntax**

```
HISTOGRAM numeric-column | expression /options;
```

**Summary of Optional Arguments**

**Appearance options**

- **BINSTART=** *number*
  
specifies the X coordinate of the first bin.

- **BINWIDTH=** *positive-number*
  
specifies the bin width.

- **BOUNDARY=** *UPPER | LOWER*
  
specifies how a boundary is counted when it lies on the endpoint of a bin.

- **DATASKIN=** *NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN*
  
enhances the visual appearance of the filled bars.

- **DATATRANSPARENCY=** *number*
  
specifies the degree of the transparency of the bar fills and bar outlines.

- **DISPLAY=** *(display-options) | STANDARD | ALL*
  
specifies which bar features to display.

- **FILLATTRS=** *(style-element | style-element(fill-options) | (fill-options)*
  
specifies the appearance of the interior fill area of the bars.

- **FILLENDCOLOR=** *color*
  
specifies the end color of the color gradient for the bar fill.

- **FILLPATTERNATTRS=** *(style-element | (fill-pattern-options)*
  
specifies the appearance of the pattern-filled areas.

- **FILLTYPE=** *(SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT)*
  
specifies the bar fill type.

- **NBINS=** *positive-integer*
  
specifies the number of bins.
ORIENT=VERTICAL | HORIZONTAL
    specifies the orientation of the Y axis and the bars.
OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
    specifies the line properties of the bar outlines.
XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
    specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Axes options
BINAXIS=TRUE | FALSE
    specifies whether to use bins as the basis for the axis tick marks.
PRIMARY=TRUE | FALSE
    specifies that the data columns for this plot and the plot type be used for determining default axis features.
SCALE=PERCENT | COUNT | PROPORTION | DENSITY
    specifies the scale for the Y axis.
XAXIS=X | X2
    specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
YAXIS=Y | Y2
    specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options
TIPFORMAT=(role-format-list)
    specifies display formats for tip columns.
TIPLABEL=(role-label-list)
    specifies display labels for tip columns.

Label options
DATALABELATTRS=style-element | style-element (text-options) | (text-options)
    specifies the color and font attributes of the bar labels.
DATALABELTYPE=NONE | AUTO | COUNT | DENSITY | PERCENT | PROPORTION
    specifies the statistic to display at the end of each bar.
ENDLABELS=TRUE | FALSE
    specifies whether the axis ticks and value labels are drawn at the bin endpoints (TRUE) or at the bin midpoints (FALSE).
LEGENDLABEL="string"
    specifies a label to be used in a discrete legend for this plot.

Midpoint options
GROUP=column | discrete-attr-var | expression
    creates a separate bar segment or bar for each unique group value in the specified column.
INCLUDEMISSINGGROUP=TRUE | FALSE
    specifies whether missing values of the group variable are included in the plot.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Statistics options

**FREQ=numeric-column | expression**
specifies a numeric column that provides frequencies for each observation that is read.

**WEIGHT=numeric-column | expression**
specifies a column that contains a bin-width calculation *a priori* weight for each observation of the input data object.

Required Argument

**numeric-column | expression**
specifies a column that contains numeric values, or a dynamic variable that refers to such a column.

Optional Arguments

**BINAXIS=TRUE | FALSE**
specifies whether to use bins as the basis for the axis tick marks.

**TRUE**
specifies that the ENDLABELS= option determines how the axis ticks and value labels are displayed.

**FALSE**
specifies that standard axes are used. Bin boundaries and midpoints that are set by the ENDLABELS= option are ignored.

Default: TRUE

Interactions
This option is ignored when this histogram is not the primary plot. For more information about primary plots, see “When Plots Share Data and a Common Axis” on page 979.

BINAXIS=TRUE is ignored when the histogram is placed on a log axis.

When this option is set to TRUE, some X-axis options that are set on the parent layout might not apply, such as INTEGER=, TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

See

“ENDLABELS=TRUE | FALSE” on page 556

“boolean ” on page 1479 for other Boolean values that you can use.

**BINSTART=number**
specifies the X coordinate of the first bin. This option can be used with the BINWIDTH= and NBINS= options to specify the bins.

Default: Determined by the system.

Interaction
This option is ignored if the specified number is greater than the maximum data value. In that case, the default starting bin is used instead.
Note: If the BINSTART= value is less than the minimum data value, then the lower end of the histogram might be padded with zero-height bins in order to accommodate the BINSTART= value.

See: XVALUES=

**BINWIDTH=** *positive-number*

specifies the bin width. This option can be used with the BINSTART= and NBINS= options to specify the bins.

Default: Determined by the system.

Restriction: Starting with SAS 9.4M1, the number of bins is limited to approximately 10,000. If the number of bins computed from the data, the BINSTART= value, and the BINWIDTH= value exceeds the limit, SAS computes a new bin-width value that yields approximately the maximum number of bins. A warning of the change is written to the SAS log in that case.

Interaction: This option is ignored when the NBINS= option is also specified (with or without the BINSTART= option) and the resulting data range does not completely span the unbinned input data range. In that case, the NBINS= option is honored and the default bin width is used instead.

Note: When BINSTART=, NBINS=, or both are specified with this option and the resulting data range is greater than the unbinned input data range, the histogram might be padded with zero-height bins in order to accommodate the BINSTART=, NBINS=, and BINWIDTH= option values.

Tip: Starting with SAS 9.4M4, the limit on the number of bins is specified by the ODS GRAPHICS statement NBINSMAX= option. The default value is 10,000. See **NBINSMAX=** on page 1494.

**BOUNDARY=** *UPPER | LOWER*

specifies how a boundary is counted when it lies on the endpoint of a bin. If this option is set to UPPER, then the value is counted as one of the values in the upper bin (the bin to the right). Otherwise, it is counted in the lower bin.

Default: **UPPER**

**DATALABELATTRS=** *style-element | style-element (text-options) | (text-options)*

specifies the color and font attributes of the bar labels.

Note: This feature applies to SAS 9.4M5 and to later releases.

Default: The GraphDataText style element

Interactions: The DATALABEL= option must be specified for this option to have any effect.

When text options are specified, any font properties that are not specified (color, family, size, weight, and style) are derived from the GraphDataText style element.
See “General Syntax for Attribute Options” on page 1519

“Text Options” on page 1525

**DATALABELTYPE=NONE | AUTO | COUNT | DENSITY | PERCENT | PROPORTION**
specifies the statistic to display at the end of each bar.

- **NONE** suppresses the data labels.
- **AUTO** uses the SCALE= option value. By default, SCALE=PERCENT.

  See “SCALE=PERCENT | COUNT | PROPORTION | DENSITY” on page 562

- **COUNT**
- **DENSITY**
- **PERCENT**
- **PROPORTION**
specifies that the count, density, percentage, or proportion statistic is to be displayed at the end of each bar.

**Default** NONE

**Interaction** When DATALABELTYPE=AUTO, the SCALE= option determines the statistic that is displayed at the end of each bar.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**
enhances the visual appearance of the filled bars. The following figure shows histogram bars with each of the skins applied.

![Histogram bars with different skins](image)

**Default** The DATASKIN= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

**Restriction** Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the
DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

**Requirement**
For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

**Interactions**
This option overrides the BEGINGRAPH statement DATASKIN= option.

The data skin appearance is based on the FILLATTRS= color.

When a data skin is applied, all bar outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

When FILLTYPE=GRADIENT is in effect, DATASKIN=SHEEN is ignored. In that case, use one of the other skins.

**DATATRANSPERENCY=** *number*

specifies the degree of the transparency of the bar fills and bar outlines.

**Default** 0

**Range** 0–1, where 0 is opaque and 1 is entirely transparent

**Tip** The FILLATTRS= option can be used to set transparency for just the bar fills. You can combine this option with FILLATTRS= to set one transparency for the bar outlines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**DISPLAY=(** *display-options*) | STANDARD | ALL**

specifies which bar features to display.

*(display-options)*
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**FILL**
displays the bar fill color

**FILLPATTERN**
displays the bar fill pattern

**Note** This option is valid starting in **SAS 9.4M1**.

**OUTLINE**
displays the bar outline

**STANDARD**
specifies FILL and OUTLINE

**ALL**
specifies all features: FILL, FILLPATTERN, and OUTLINE

**Default** The GraphHistogram:DisplayOpts style reference.

**Tip** Use the OUTLINEATTRS=, FILLATTRS=, and FILLPATTERNATTRS= options to control the appearance of the bars.
**ENDLABELS=** TRUE | FALSE

Specifies whether the axis ticks and value labels are drawn at the bin endpoints (TRUE) or at the bin midpoints (FALSE).

**Default** FALSE.

**Interactions**
This option is ignored if this plot is not the primary plot in the parent layout. For more information, see the PRIMARY= option.

This option is ignored if BINAXIS= FALSE. By default, BINAXIS=TRUE.

If the TICKS= suboption is specified in the XAXISOPTS= option, then this option is ignored.

**See** “boolean ” on page 1479 for other Boolean values that you can use.

**FILLATTRS=** style-element | style-element (fill-options) | (fill-options)

Specifies the appearance of the interior fill area of the bars.

**Defaults**
In SAS 9.4M1 and earlier releases, the Color attribute of the GraphDataDefault style element

Starting with SAS 9.4M2, the Color attribute of the GraphDataDefault style element for non-grouped data or the GraphData1–GraphDataN style elements for grouped data.

**Interactions**
For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

When a discrete attribute map is in effect for the histogram, the fill colors that are specified by the attribute map take precedence over COLOR. If the attribute map provides no color for a value, COLOR prevails for that value.

**Tip**
The DATATRANSPARENCY= option sets the transparency for bar fills and bar outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the fills. Example:

datatransparency=0.2 fillattrs=(transparency=0.6)

**See** “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

**FILLENDCOLOR=** color

Specifies the end color of the color gradient for the bar fill.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Default** the graph wall background color, even when the wall is not displayed

**Requirement** FILLTYPE=COLORGRADIENT must be in effect. Otherwise, this option is ignored.
FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

Note: This option is valid starting in SAS 9.4M1.

Defaults
For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.
For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

Restriction
In SAS 9.4M4 and in earlier releases, the only styles that are delivered by SAS that support fill patterns are JOURNAL2, JOURNAL3, and MONOCHROMEPRINTER. If style-element is specified and the active ODS style does not support fill patterns, this option is ignored.

Interaction
For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

Tip
In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=None in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
“Fill Pattern Options” on page 1521 for available fill-pattern-options.

FILLTYPE=SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT
specifies the bar fill type.

Note: This feature applies to SAS 9.4M2 and to later releases.

SOLID
fills each bar with the color that is assigned to that bar.

ALPHAGRADIENT
fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.

Note: This feature applies to SAS 9.4M5 and to later releases.

Tips
The fill color is determined by a style element or by the FILLATTRS= option COLOR= suboption. The initial transparency is determined by the DATATRANSPARENCY= option or by the FILLATTRS= option TRANSPARENCY= suboption.
For grouped plots, use the FILLATTRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**COLORGRADIENT**

fills each bar with a color gradient that starts with the fill color that is assigned to that bar and ends with the color that is specified in the FILLENDCOLOR= option.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Alias**

GRADIENT

**Interaction**

The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

**Tip**

Use the DATATRANSPARENCY= option or the FILLATTRS= option TRANSPARENCY= suboption to set the initial transparency in the gradients.

**See**

“FILLENDCOLOR=color” on page 556

**GRADIENT**

fills each bar with a color and a transparency gradient that starts at the bar top with the initial transparency that is assigned to that bar, and ends with full transparency at the bar baseline.

**Note:** This option applies to SAS 9.4M2 through SAS 9.4M4. Starting with SAS 9.4M5, ALPHAGRADIENT replaces GRADIENT, and GRADIENT is changed to an alias of COLORGRADIENT.

**Interaction**

The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

**Tips**

The initial fill color is determined by a style element or by the FILLATTRS= option COLOR= suboption. The initial transparency is determined by the DATATRANSPARENCY= option or by the FILLATTRS= option TRANSPARENCY= suboption.

For grouped plots, use the FILLATTRS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**Default**

SOLID

**Interaction**

The DISPLAY= option must include FILL for this option to have any effect.

**Note**

The output for SAS programs written prior to SAS 9.4M5 that specify FILLTYPE=GRADIENT might change starting with SAS 9.4M5. To restore the original appearance, specify FILLTYPE=ALPHAGRADIENT.
See \textit{DATASKIN=} on page 554

**FREQ=**\textit{numeric-column | expression}
specifies a numeric column that provides frequencies for each observation that is read.

- Default: All observations have a frequency count of 1.
- Restriction: If the value of the \textit{numeric-column} is missing or is less than 1, then the observation is not used in the analysis. If the value is not an integer, then only the integer portion is used.
- Note: If \( n \) is the value of the numeric column for a given observation, then that observation is used \( n \) times for the purposes of any statistical computation.

**GROUP=**\textit{column | discrete-attr-var | expression}
creates a separate bar segment or bar for each unique group value in the specified column.

Note: This feature applies to \textit{SAS 9.4M2} and to later releases.

\textit{discrete-attr-var}
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a \textit{DISCRETEATTRVAR} statement.

- Restriction: A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

A distinct bar or bar segment is created for each group value by varying the visual attributes of the bar display features that are enabled by the current ODS style or by the \textit{DISPLAY=} option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature(^1)</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1--GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern</td>
<td>FillPattern and ContrastColor attributes of a GraphData1--GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern. If the active ODS style does not specify FillPattern, fill patterns are not displayed by default.</td>
</tr>
<tr>
<td>Display Feature</td>
<td>Style Attributes That Control Default Group Appearance</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>Outline</td>
<td>- if the outline is enabled with fill color, ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style</td>
</tr>
<tr>
<td></td>
<td>- if the outline is enabled with fill pattern or if the outline is the only display feature enabled, ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element</td>
</tr>
</tbody>
</table>

1 The default display features are determined by the DisplayOpt attribute of the GraphHistogram style element.

**Note**
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tip**
You can individually override the representations that are used to identify the groups. For example, in some ODS styles, each distinct group value is represented by a different line pattern for the bar outlines. In that case, you can use the PATTERN= setting on the OUTLINEATTRS= option to assign the same line pattern to all of the bar outlines.

**See**
"Example 2: Histogram of Highway Mileage Grouped by Origin" on page 565
"DISCRETEATTRVAR Statement" on page 1435

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**Default** TRUE

**Interaction** For this option to take effect, the GROUP= option must also be specified.

**Tip**
The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.
See “boolean ” on page 1479 for other Boolean values that you can use.

**LEGENDLABEL=** *string*

specifies a label to be used in a discrete legend for this plot.

Default The *string* specified on the NAME= option.

Restriction This option applies only to an associated DISCRETELEGEND statement.

**NAME=** *string*

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**NBINS=** *positive-integer*

specifies the number of bins. This option can be used with the BINSTART= and BINWIDTH= options to specify the bins.

Default Determined by the system.

Ranges A positive integer in SAS 9.4 and earlier releases

2–10000 starting with SAS 9.4M1

Note When BINWIDTH=, BINSTART=, or both are specified with this option and the resulting data range is greater than the unbinned input data range, the histogram might be padded with zero-height bins in order to accommodate the BINSTART=, BINWIDTH=, and NBINS= option values.

Tip Starting with SAS 9.4M4, the limit on the number of bins is specified by the ODS GRAPHICS statement NBINSMAX= option. The default value is 10,000. See NBINSMAX= on page 1494.

**ORIENT=** VERTICAL | HORIZONTAL

specifies the orientation of the Y axis and the bars.

Default VERTICAL

**OUTLINEATTRS=** *style-element* | *style-element* *(line-options)* | *(line-options)*

specifies the line properties of the bar outlines.

Defaults For non-grouped data, the GraphOutlines style element.

In SAS 9.4M1 and earlier releases, for grouped data and filled bars, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the LineThickness attribute of the GraphOutlines style element.
Starting with SAS 9.4M2, for grouped data and filled bars, the ContrastColor attribute of the GraphData1–GraphDataN style elements, and the LineThickness and LineStyle attributes of the GraphOutlines style element.

**Interaction**
For this option to have any effect, the outlines must be enabled by the ODS style or by the DISPLAY= option.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element.

“Line Options” on page 1522 for available line-options.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**
FALSE

**Restriction**
This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note**
In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

**See**
“When Plots Share Data and a Common Axis” on page 979

“boolean” on page 1479 for other Boolean values that you can use.

**SCALE=PERCENT | COUNT | PROPORTION | DENSITY**
specifies the scale for the Y axis.

**PERCENT**
displays the percentages between 0 and 100 on the Y axis.

**COUNT**
displays the frequency counts on the Y axis.

**PROPORTION**
displays the proportions between 0 and 1 on the Y axis.

**DENSITY**
displays the true density estimates on the Y axis.

**Default**
PERCENT

**Requirement**
If you need to include a histogram and a density plot in your graph and both plots specify the same data column, place the HISTOGRAM statement before the DENSITYPLOT statement, or specify SCALE=DENSITY in the HISTOGRAM statement. Otherwise, the density plot values might not scale properly to the histogram bins on the dependent axis. This requirement applies even when the histogram and density plot are in different cells in the graph.
Tip When SCALE=PERCENT, the response axis tick values do not include a percent sign after each value. To add a percent sign after each response axis tick value in that case, use SCALE=PROPORTION, and then specify the options shown in the following example for the response axis:

\[
yaxisopts=(label="Percent" linearopts=(tickvalueformat=percent.))
\]

**TIPFORMAT=(role-format-list)**

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

- **role-format-list**
  - a space-separated list of role-name = format pairs.

  **Example**
  ```
  TIPFORMAT=(Y=6.2)
  ```

  **Default**
  The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

  **Requirement**
  To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

  **Note**
  The X role represents the binned value. The Y role represents the computed amount of X in the units specified by the SCALE= option.

**TIPLABEL=(role-label-list)**

specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

- **role-label-list**
  - a space-separated list of rolename = "string" pairs.

  **Example**
  ```
  TIPLABEL=(Y="Percent")
  ```

  **Default**
  The column label or column name of the column assigned to the role.

  **Note**
  The Y role represents the computed amount of X in the units specified by the SCALE= option.

**WEIGHT=numeric-column | expression**

specifies a column that contains a bin-width calculation a priori weight for each observation of the input data object.

  **Requirement**
  The value must be nonnegative.

  **Note**
  The values of the weight variable must be greater than zero. If the weight value for an observation is zero, negative, or missing, that observation is deleted from the analysis.

**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

  **Default**
  X
Interaction  The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS**

specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins. For example, if BINSTART=10 and BINWIDTH=10, then using LEFTPOINTS would result in bins 10 - 20, 20 - 30, and so on. Using RIGHTPOINTS would result in bins 0 - 10, 10 - 20, ...., and using MIDPOINTS would result in bins 5 - 15, 15 - 25, ... .

Default  MIDPOINTS

**YAXIS=Y | Y2**  
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default  Y

Interaction  The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details

The histogram’s default bin width is computed by using the number of observations and the range of the data. When a curve is overlaid on the histogram, the histogram’s bin width is used to scale the curve so that the area under the curve is equal to the area of the histogram.

The X axis and Y axis are linear by default. You can change axis properties with the XAXISOPTS= and YAXISOPTS= options of the layout statement.

Examples:

Example 1: Histogram of Vehicle Weight

The following graph was generated by the “Example Program” on page 565:
Example Program

```sas
proc template;
define statgraph histogram;
begingraph;
   entrytitle "Histogram of Vehicle Weights";
   layout overlay /
      xaxisopts=(label="Vehicle Weight (LBS)");
      yaxisopts=(griddisplay=on);
      histogram weight;
   endlayout;
endgraph;
end;
run;

proc sgrender data=sashelp.cars template=histogram;
run;
```

Example 2: Histogram of Highway Mileage Grouped by Origin

The following graph was generated by the “Example Program” on page 566.
Example Program

```sas
proc template;
    define statgraph histogram;
    begingraph;
        entrytitle "Highway Mileage Distribution by Origin";
        layout overlay /;
            histogram mpg_highway / name="histogram" group=origin;
            discretelegend "histogram" / title="Origin:";
        endlayout;
    endgraph;
end;
run;

proc sgrender template=histogram data=sashelp.cars;
run;
```

HISTOGRAMPARAM Statement

Creates a univariate histogram for specified values of bin midpoints and bin frequencies.

Restrictions:
- Only uniform width bins are supported.
- Starting with SAS 9.4M4, the number of bins is limited to 10,000. If the number of bins exceeds the limit, the histogram is not drawn. A warning message is written to the SAS log in that case.
Tips: Starting with SAS 9.4M4, the limit on the number of bins is specified by the ODS GRAPHICS statement NBINSMAX= option. The default value is 10,000. See NBINSMAX= on page 1494.

Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

HISTOGRAMPARM X=numeric-column | expression
Y=non-negative-numeric-column | expression <options>;

Summary of Optional Arguments

Appearance options

  DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of the filled bars.

  DATATRANSPARENCY=number
  specifies the degree of the transparency of the bars (outline and fill).

  DISPLAY=(display-options) | STANDARD | ALL
  specifies which bar features to display.

  FILLATTRS=style-element | style-element (fill-options) | (fill-options)
  specifies the appearance of the interior fill area of the bars.

  FILLENDCOLOR=color
  specifies the end color of the color gradient for the bar fill.

  FILLPATTERNATTRS=style-element | (fill-pattern-options)
  specifies the appearance of the pattern-filled areas.

  FILLTYPE=SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT
  specifies the bar fill type.

  ORIENT=VERTICAL | HORIZONTAL
  specifies the orientation of the Y axis and the bars.

  OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
  specifies the line properties of the bar outlines.

  XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS
  specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Axes options

  BINAXIS=TRUE | FALSE
  specifies whether to use bins as the basis for the axis tick marks.

  PRIMARY=TRUE | FALSE
  specifies that the data columns for this plot and the plot type be used for determining default axis features.

  XAXIS=X | X2
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

  YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

ROLENNAME=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a histogram bin.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns.

TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options

DATALABEL=column | expression
specifies a column for the bar labels.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the bar labels.

DATALABELFITPOLICY=AUTO | NONE | ROTATE | SPLIT | SPLITALWAYS
specifies a policy for avoiding collisions among the bar labels when labels are displayed.

DATALABELSPLITCHAR="character-list"
specifies one or more characters on which the data labels can be split.

DATALABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the data labels.

ENDLABELS=TRUE | FALSE
specifies whether the axis ticks and value labels are drawn at the bin endpoints (TRUE) or at the bin midpoints (FALSE).

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Plot reference options

NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments

X=numeric-column | expression
specifies the column for the midpoint values.

Requirement

When you plot a SAS data set, you must sort the data in ascending order of the specified X column before you plot the data. When you plot a CAS in-memory table, you do not need to sort the data. The data is automatically sorted in ascending order of the X column formatted character values or of the unformatted numeric values before the data is plotted.

Y=non-negative-numeric-column | expression
specifies the column for the frequency values. These values can be frequency counts, percentages, or proportions between 0 and 1.
Requirement: The Y values cannot be negative.

Optional Arguments

**BINAXIS=TRUE | FALSE**

specifies whether to use bins as the basis for the axis tick marks.

- **TRUE**
  specifies that the ENDLABELS= option determines how the axis ticks and value labels are displayed.

- **FALSE**
  specifies that standard axes are used. Bin boundaries and midpoints that are set by the ENDLABELS= option are ignored.

Default: **TRUE**

Interactions

This option is ignored when this histogram is not the primary plot. For more information about primary plots, see "When Plots Share Data and a Common Axis" on page 979.

BINAXIS=TRUE is ignored when the histogram is placed on a log axis.

When this option is set to TRUE, some X-axis options that are set on the parent layout might not apply, such as INTEGER=, TICKVALUELIST=, TICKVALUESEQUENCE=, and INCLUDERANGES=.

See **"ENDLABELS=TRUE | FALSE" on page 575**

"boolean " on page 1479 for other Boolean values that you can use.

**DATALABEL=column | expression**

specifies a column for the bar labels. The labels appear at the top or end of each bar, depending on the chart orientation.

Default: No data labels are displayed

**DATALABELATTRS=style-element | style-element (text-options) | (text-options)**

specifies the color and font attributes of the bar labels.

Default: The GraphDataText style element

Interactions

The DATALABEL= option must be specified for this option to have any effect.

When text options are specified, any font properties that are not specified (color, family, size, weight, and style) are derived from the GraphDataText style element.

See **“General Syntax for Attribute Options” on page 1519**

"Text Options" on page 1525
DATALABELFITPOLICY= AUTO | NONE | ROTATE | SPLIT | SPLITALWAYS
specifies a policy for avoiding collisions among the bar labels when labels are displayed.

**AUTO**
selects a collision avoidance policy based on the chart orientation and data type.

For a numeric column with ORIENT=VERTICAL, AUTO rotates the labels if they do not fit the midpoint spacing. For a character column, AUTO splits the labels if they do not fit the midpoint spacing.

**Note** When ORIENT=HORIZONTAL, AUTO always draws the labels horizontally.

**Tip** If character labels do not fit after splitting, then try using ROTATE instead of AUTO.

**See** ORIENT= on page 578 for information about chart orientation.

**NONE**
does not attempt to fit bar labels that collide.

**ROTATE**
rotates the bar labels for vertical bars if the labels collide in the available width.

**Requirement** The chart orientation must be vertical (ORIENT=VERTICAL).

**SPLIT**
splits the label for vertical bars at a split character only if a split is needed at that character in order to make the label fit the available space.

No split occurs at split characters that occur where a split is not needed. If the label does not contain any of the specified split characters, then a split does not occur. In that case, if the label does not fit the available space, then it might collide with the adjoining labels.

**Requirement** The chart orientation must be vertical (ORIENT=VERTICAL).

**See** DATALABELSPLITCHAR= for information about specifying the split characters.

**SPLITALWAYS**
splits the label for vertical bars at every occurrence of a split character.

If the label does not contain any of the specified split characters, then a split does not occur.

**Requirement** The chart orientation must be vertical (ORIENT=VERTICAL).

**See** DATALABELSPLITCHAR= for information about specifying the split characters.

**INSIDEPREFERRED**
Attempts to place the data labels in a horizontal bar chart inside the bars.

**Note:** This feature applies to SAS 9.4M5 and to later releases.
Restrictions
This option applies to non-grouped or cluster-grouped horizontal bar charts only.

Label splitting is not supported when a label is placed inside a bar.

Requirement
For a grouped horizontal bar chart, the GROUPDISPLAY=CLUSTER option must be in effect. Otherwise, the INSIDEPREFERRED policy is ignored.

Interaction
If SEGMENTLABEL=TRUE is in effect, the INSIDEPREFERRED policy is ignored, and the labels are placed outside.

Notes
If any bar label does not fit inside its bar because the font height is too high, all of the bar labels are placed outside.

If any bar label cannot fit inside the length of its bar, that label is placed outside of its bar and is fit using the AUTO fit policy.

If the outside labels collide because the font height is greater than the midpoint space, then all labels are dropped.

WARNING: The bar labels are suppressed. Use DATALABELFITPOLICY=NONE to force the labels to be displayed.

Default
AUTO

Requirement
The DATALABEL= option must also be specified.

DATALABELSPLITCHAR="character-list"

specifies one or more characters on which the data labels can be split. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABELFITPOLICY=SPLIT, data labels are split on a split character only if a split is needed at that point in order to make the label fit. When DATALABELFITPOLICY=SPLITALWAYS, the data labels are split unconditionally on each occurrence of a split character. If the data label does not contain any of the specified characters, then the label is not split.

"character-list"
one or more characters with no space between each character and enclosed in quotation marks.

Default
A blank space

Requirements
The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

datalabelsplitsplitchar="abc"

The DATALABELFITPOLICY= option must specify SPLIT or SPLITALWAYS.
Interactions

The DATALABELFITPOLICY= option specifies the policy that is used to manage the split behavior of the data labels.

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

Notes

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels.

**TRUE**

drops a split character from the data label when a split occurs at that character.

Split characters at which a split does not occur are left in place. The DATALABELFITPOLICY= option determines where the labels are split. When DATALABELFITPOLICY=SPLIT, each label is split at a split character only where a split is needed in order to make the label fit the available space. At each split point, the split character is dropped, and the characters that follow the split character, up to but not including the split character at the next split point, are wrapped to the following line.

When DATALABELFITPOLICY=SPLITALWAYS, each label is split at every instance of a split character. All of the split characters are dropped. The characters that follow each split character, up to but not including the next split character, are wrapped to the next line.

The following figure shows how label “Product*Group*1” is split when the DATALABELSPLITCHARDROP=TRUE and DATALABELSPLITCHAR=“*” options are used with the SPLIT and SPLITALWAYS fit policies.

![Figure showing how label “Product*Group*1” is split](image)

In this example, when DATALABELFITPOLICY=SPLIT, the label is split at the first occurrence of the asterisk in order to make the label fit. No split is needed at the second asterisk. The first asterisk is dropped, and Group*1 wraps to the next line. Notice that the second asterisk is not dropped in this case. When DATALABELFITPOLICY=SPLITALWAYS, the label is split at every occurrence of the asterisk. In this case, both asterisks are dropped, and the characters that follow each asterisk wrap to the next line.

**FALSE**

includes the split characters in the data label.

The DATALABELFITPOLICY= option determines how the split characters are displayed. When DATALABELFITPOLICY=SPLIT, each data label is split at a split character only where a split is needed in order to make the label fit the available space. A split might not occur at every split character in the label. At each split point, the split character remains as the last character in the current line. The characters that follow the split character, up to and including
the split character at the next split point, are then wrapped to the following line. This process repeats until the entire data label is displayed.

When DATALABELFITPOLICY=SPLIT, each data label is split at every instance of a split character in the label regardless of whether a split is actually needed. Each split character remains as the last character in the current line. The characters that follow each split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows how label “Product*Group*1” is split when the DATALABELSPLITCHARDROP=FALSE and DATALABELSPLITCHAR=”*” options are used with the SPLIT and SPLITALWAYS fit policies.

<table>
<thead>
<tr>
<th>DATALABELFITPOLICY=SPLIT</th>
<th>Product* Group*1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATALABELFITPOLICY=SPLIT</td>
<td>Product* Group*1</td>
</tr>
<tr>
<td></td>
<td>Group* 1</td>
</tr>
</tbody>
</table>

In this example, when DATALABELFITPOLICY=SPLIT, the label is split at the first occurrence of the asterisk in order to make the label fit. No split is needed at the second asterisk. The characters that follow the first asterisk wrap to the next line. When DATALABELFITPOLICY=SPLIT, the label is split at every occurrence of the asterisk. Each asterisk remains as the last character in the current line, and the characters that follow are wrapped to the next line.

Default: TRUE. A split character is dropped from the data label when a split occurs at that character.

Requirements: The DATALABEL= option must also be specified.

The DATALABELFITPOLICY= option must specify SPLIT or SPLITALWAYS.

Interaction: The DATALABELSPLITCHAR= option specifies the split characters.

See: “boolean ” on page 1479 for other Boolean values that you can use.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the filled bars. The following figure shows histogram bars with each of the skins applied.
The `DATASKIN=` option value that is specified in the `BEGINGRAPH` statement. If that value is not specified, then the `GraphSkins:DataSkin` style element value is used.

**Restriction**
Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an `OVERLAY` or `PROTOTYPE` layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the `DATASKINMAX=` option in your `ODS GRAPHICS` statement to increase the maximum limit.

**Requirement**
For this option to have any effect, the fill must be enabled by the ODS style or the `DISPLAY=` option.

**Interactions**
This option overrides the `BEGINGRAPH` statement `DATASKIN=` option.

The data skin appearance is based on the `FILLATTRS=` color.

When a data skin is applied, all bar outlines are set by the skin, and the `OUTLINEATTRS=` option is ignored.

When `FILLTYPE=GRADIENT` is in effect, `DATASKIN=SHEEN` is ignored. In that case, use one of the other skins.

**DATATRANSPARENCY=** `number`
specifies the degree of the transparency of the bars (outline and fill).

**Default**
0

**Range**
0−1, where 0 is opaque and 1 is entirely transparent

**Tip**
The `FILLATTRS=` option can be used to set transparency for just the bar fills. You can combine this option with `FILLATTRS=` to set one transparency for the bar outlines but a different transparency for the bar fills. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**DISPLAY=(display-options) | STANDARD | ALL**
specifies which bar features to display.
(display-options)  
a space-separated list of one or more of the following options, enclosed in parentheses:

FILL  
displays the bar fill color

FILLPATTERN  
displays the bar fill pattern

Note  This option is valid starting in SAS 9.4M1.

OUTLINE  
displays the bar outline

STANDARD  
specifies FILL and OUTLINE

ALL  
specifies all features: FILL, FILLPATTERN, and OUTLINE

Default  The GraphHistogram:DisplayOpts style reference.

Tip  Use the OUTLINEATTRS=, FILLATTRS=, and FILLPATTERNATTRS= options to control the appearance of the bars.

ENDLABELS=TRUE | FALSE  
specifies whether the axis ticks and value labels are drawn at the bin endpoints (TRUE) or at the bin midpoints (FALSE).

Default  FALSE.

Interactions  This option is ignored if this plot is not the primary plot in the parent layout. For more information, see the PRIMARY= option.

This option is ignored if BINAXIS= FALSE. By default, BINAXIS=TRUE.

If the TICKS= suboption is specified in the XAXISOPTS= option, then this option is ignored.

See  "boolean " on page 1479 for other Boolean values that you can use.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)  
specifies the appearance of the interior fill area of the bars.

Default  The GraphDataDefault style element

Interactions  For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

When a discrete attribute map is in effect for the histogram, the fill colors that are specified by the attribute map take precedence over COLOR. If the attribute map provides no color for a value, COLOR prevails for that value.

Tip  The DATATRANSPARENCY= option sets the transparency for bar fills and bar outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines but a different transparency for the fills. Example:
FILLENDCOLOR=color
specifies the end color of the color gradient for the bar fill.

Note: This feature applies to SAS 9.4M5 and to later releases.

Default the graph wall background color, even when the wall is not displayed

Requirement FILLTYPE(COLORGRADIENT) must be in effect. Otherwise, this option is ignored.

See "color " on page 1480

FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

Note: This option is valid starting in SAS 9.4M1.

Defaults For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.

For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

Interaction For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.

Tip In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see "Attribute Rotation Patterns" in SAS Graph Template Language: User’s Guide.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Pattern Options” on page 1521 for available fill-pattern-options.

FILLTYPE=SOLID | ALPHAGRADIENT | COLORGRADIENT | GRADIENT
specifies the bar fill type.

Note: This feature applies to SAS 9.4M2 and to later releases.

SOLID
fills each bar with the color that is assigned to that bar.
ALPHAGRADIENT
fills each bar with a color and a transparency gradient that starts at the bar
top with the initial transparency that is assigned to that bar, and ends with full
transparency at the bar baseline.

Note: This feature applies to SAS 9.4M5 and to later releases.

Tips
The fill color is determined by a style element or by the FILLATTRS=
option COLOR= suboption. The initial transparency is determined by
the DATATRANSPARENCY= option or by the FILLATTRS= option
TRANSPARENCY= suboption.

For grouped plots, use the FILLATTRS= option in a discrete attribute
map to set the initial transparency in the gradients for specific values.

COLORGRADIENT
fills each bar with a color gradient that starts with the fill color that is assigned
to that bar and ends with the color that is specified in the FILLENDCOLOR= option.

Note: This feature applies to SAS 9.4M5 and to later releases.

Alias
GRADIENT

Interaction
The SHEEN data skin cannot be used when
FILLTYPE=GRADIENT is in effect. You can use one of the other
data skins.

Tip
Use the DATATRANSPARENCY= option or the FILLATTRS=
option TRANSPARENCY= suboption to set the initial
transparency in the gradients.

See
“FILLENDCOLOR=color” on page 576

GRADIENT
fills each bar with a color and a transparency gradient that starts at the bar
top with the initial transparency that is assigned to that bar, and ends with full
transparency at the bar baseline.

Note: This option applies to SAS 9.4M2 through SAS 9.4M4. Starting with
SAS 9.4M5, ALPHAGRADIENT replaces GRADIENT, and GRADIENT is
changed to an alias of COLORGRADIENT.

Interaction
The SHEEN data skin cannot be used when
FILLTYPE=GRADIENT is in effect. You can use one of the other
data skins.

Tips
The initial fill color is determined by a style element or by the
FILLATTRS= option COLOR= suboption. The initial transparency
is determined by the DATATRANSPARENCY= option or by the
FILLATTRS= option TRANSPARENCY= suboption.
For grouped plots, use the FILLATRGS= option in a discrete attribute map to set the initial transparency in the gradients for specific values.

**Default**

SOLID

**Interaction**

The DISPLAY= option must include FILL for this option to have any effect.

**Note**

The output for SAS programs written prior to SAS 9.4M5 that specify FILLTYPE=GRADIENT might change starting with SAS 9.4M5. To restore the original appearance, specify FILLTYPE=ALPHAGRADIENT.

**See**

DATASKIN= on page 573

**LEGENDLABEL="string"**

specifies a label to be used in a discrete legend for this plot.

**Default**

The *string* specified on the NAME= option.

**Restriction**

This option applies only to an associated DISCRETELEGEND statement.

**NAME="string"**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**

The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**

The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**ORIENT=VERTICAL | HORIZONTAL**

specifies the orientation of the Y axis and the bars.

**Default**

VERTICAL

**OUTLINEATRGS=style-element | style-element (line-options) | (line-options)**

specifies the line properties of the bar outlines.

**Default**

The GraphOutlines style element.

**Interaction**

For this option to have any effect, the outlines must be enabled by the ODS style or the DISPLAY= option.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.
### PRIMARY=TRUE | FALSE

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

- **Default**: FALSE
- **Restriction**: This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.
- **Note**: In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.
- **See**: “When Plots Share Data and a Common Axis” on page 979

### ROLENAME=(role-name-list)

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

- **(role-name-list)**: a space-separated list of role-name = column pairs.
- **Example**: The following example assigns the column Obs to the user-defined role TIP:
  - `ROLENAME=(TIP1=OBS)`
- **Default**: No user-defined roles
- **Requirement**: The role names that you choose must be unique and different from the predefined roles X, Y, and DATALABEL.

### TIP=(role-list) | NONE

specifies the information to display when the cursor is positioned over a histogram bin. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the histogram can be specified along with roles that do.

- **(role-list)**: an ordered, space-separated list of unique HISTOGRAMPARM and user-defined roles.
- **Tip**: User-defined roles are defined with the ROLENAME= option.
- **Example**: The following example displays data tips for the columns assigned to the roles X and Y, as well as the column Pct, which is not assigned to any pre-defined HISTOGRAMPARM role. The Pct column must first be assigned a role:
  - `ROLENAME=(TIP1=PCT)`
  - `TIP=(X Y TIP1)`
NONE
   suppresses data tips from the plot.

Default
   The columns assigned to these roles are automatically included in
   the data tip information: X and Y.

Requirement
   To enable data tips in the output, you must include an ODS
   GRAPHICS ON statement that specifies the IMAGEMAP option,
   and you must write the output to the ODS HTML destination.

Interaction
   This option is ignored when the plot statement is in an OVERLAY
   or PROTOTYPE layout and the INCLUDERANGES= option is
   specified in the LINEAROPTS= or TIMEOPTS= option for either
   axis.

Tip
   The labels and formats for the TIP roles can be controlled with the
   TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
   specifies display formats for tip columns. This option provides a way to control
   the formats of columns that appear in data tips.

(role-format-list)
   a space-separated list of role-name = format pairs.

Example
   ROLENAME=(TIP1=Salary)
   TIP=(TIP1)
   TIPFORMAT=(TIP1=Dollar12.)

Default
   The column format of the column assigned to the role or BEST6 if
   no format is assigned to a numeric column.

Restriction
   Only the roles that appear in the TIP= option are used.

Requirement
   A column must be assigned to each of the specified roles. (See the
   ROLENAME= option.)

TIPLABEL=(role-label-list)
   specifies display labels for tip columns. This option provides a way to control
   the labels of columns that appear in data tips.

role-label-list
   a space-separated list of rolename = "string" pairs.

Example
   ROLENAME=(TIP1=PCT)
   TIP=(TIP1)
   TIPLABEL=(TIP1="Percent")

Default
   The column label or column name of the column assigned to the
   role.

Restriction
   Only the roles that appear in the TIP= option are used.

Requirement
   A column must be assigned to each of the specified roles. (See the
   ROLENAME= option.)
**XAXIS=X | X2**  
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

**XVALUES=MIDPOINTS | LEFTPOINTS | RIGHTPOINTS**  
specifies whether the input X values represent midpoints, lower endpoints, or upper endpoints of the bins.

Default: MIDPOINTS

**YAXIS=Y | Y2**  
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details

When a curve is overlaid on the histogram, the histogram bin width is used to scale the curve so that the area under the curve is equal to the area of the histogram.

The X axis and Y axis are linear by default. You can change axis properties with the XAXISOPTS= and YAXISOPTS= options of the LAYOUT OVERLAY statement.

Example: HISTOGRAMPARM Statement

The following graph was generated by the “Example Program” on page 582:
Example Program

```
proc template;
  define statgraph histogramparm;
  begingraph;
    entrytitle "Histogram of Vehicle Weights";
    layout overlay;
      histogramparm x=midpoint y=frequency;
    endlayout;
  endgraph;
end;
run;

data bins;
  input Midpoint Frequency;
  label midpoint="Vehicle Weight (LBS)";
  datalines;
    2000 18
    3000 208
    4000 159
    5000 36
    6000 6
    7000 1
  ;
run;

proc sgrender data=bins template=histogramparm;
run;
```
LINECHART Statement

Creates a line chart that is computed from input data.

Restrictions: GROUPDISPLAY= STACK has stringent requirements for the data. If the requirements are not met for all the data, then the line chart is not drawn. If the LINECHART statement is in a LAYOUT DATALATTICE or LAYOUT DATAPANEL block, then the requirements are tested for all of the data, not for each individual panel.

The category axis (the X axis when ORIENT=VERTICAL or the Y axis when ORIENT=HORIZONTAL) is always discrete.

The response axis (the Y axis when ORIENT=VERTICAL or the X axis when ORIENT=HORIZONTAL) is always numeric.

The LINECHART statement performs discrete binning for a numeric category column only.

Note: Specifying only the CATEGORY= option creates a computed line chart with vertices representing frequency counts or percents of unique CATEGORY values. For a non-grouped chart, specifying both the CATEGORY= and RESPONSE= options creates a computed line chart with vertices representing the summarized values of the RESPONSE values that are categorized by unique CATEGORY values.

Tips: The line segments in the chart always join the categorical values in the order in which they appear on the axis. By default, the vertices in the line chart appear in the order in which the X values occur in the input data. To change the categorical axis tick value order, use the SORTORDER= or TICKVALUELIST= suboption of the DISCRETEOPTS= option for the X axis. For example:

```
XAXISOPTS=(DISCRETEOPTS=(SORTORDER=ASCENDINGFORMATTED))
```

The response axis of the line chart includes the zero value by default. You can use the BASELINEINTERCEPT= to change the Y intercept.

By default, missing category values are ignored. To treat missing category values as a category, include the INCLUDEMISSINGDISCRETE=TRUE option in the BEGINGRAPH statement.

Starting with SAS 9.4M3, subpixel rendering is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

See:

- “LAYOUT DATALATTICE Statement” on page 47
- “LAYOUT DATAPANEL Statement” on page 74
- “Axis Options in Layouts” on page 987
- “BEGINGRAPH Statement” on page 23

Syntax

```
LINECHART CATEGORY=column | expression </options> ;
LINECHART CATEGORY=column | expression RESPONSE=numeric-column | expression </options> ;
```
Summary of Optional Arguments

Appearance options

**BREAK=TRUE | FALSE**
breaks the line at missing values of the RESPONSE variable.

**COLORMODEL=color-ramp-style-element | (color-list)**
specifies a color ramp to use with the COLORRESPONSE= option.

**COLORRESPONSE=numeric-column | range-attr-var | expression**
specifies the column or range attribute variable to use to map the line, marker, and fill colors.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**
enhances the visual appearance of the lines and markers.

**DATATRANSPERTRANCY=number**
specifies the degree of the transparency of the line, markers, and vertex labels, if displayed.

**DISPLAY=(display-options) | STANDARD | ALL**
specifies which line-chart features to display.

**FILLATTRS=style-element | (fill-options)**
specifies the appearance of the filled area.

**FILLEDOUTLINEDMARKERS=TRUE | FALSE**
specifies whether markers are drawn with both fill and an outline.

**GROUPDISPLAY=OVERLAY | STACK**
specifies how to display grouped lines.

**INDEX=positive-integer-column | expression**
specifies indices for mapping line attributes (color, marker symbol, and line pattern) and fill attributes to one of the GraphData1–GraphDataN style elements.

**LINEATTRS=style-element | (line-options)**
specifies the appearance of the line.

**MARKERATTRS=style-element | style-element (marker-options) | (marker-options)**
specifies the attributes of the data markers.

**MARKERFILLATTRS=style-element | (fill-options)**
specifies the appearance of the filled markers.

**MARKEROUTLINEATTRS=style-element | (line-options)**
specifies the appearance of the marker outlines.

**ORIENT=VERTICAL | HORIZONTAL**
specifies the orientation of the Y axis.

**SMOOTHCONNECT=TRUE | FALSE**
specifies that a smoothed line passes through all vertices.

Axes options

**BASELINEINTERCEPT=number | AUTO | AXISMIN | AXISMINEXTEND | AXISMAX | AXISMAXEXTEND**
specifies the Y-intercept for the baseline.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features.

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options
ROLENAME=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips.
TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a line.
TIPFORMAT=(role-format-list)
specifies display formats for tip columns.
TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options
LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.
VERTEXLABEL=TRUE | FALSE
specifies whether to label the vertices with their response value (or statistic).
VERTEXLABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the vertex labels.
VERTEXLABELFORMAT=format
specifies the format used to display the vertex label response or statistic.

Midpoint options
GROUP=column | discrete-attr-var | expression
creates a separate line for each unique group value in the specified column.
GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.
INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

ODS options
URL=string-column
specifies an HTML page to display when a line segment, marker, or fill area is selected.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Statistics options
STAT=FREQ | PCT | PROPORTION | SUM | MEAN
specifies the statistic to be computed for the Y axis.
Required Argument

**CATEGORY=column | expression**

specifies the column or expression for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

Optional Arguments

**BASELINEINTERCEPT=number | AUTO | AXISMIN | AXISMINEXTEND | AXISMAX | AXISMAXEXTEND**

specifies the Y-intercept for the baseline.

*number*

specifies the Y coordinate of the baseline.

This value is included in the data range that is reported by the line chart.

**Interaction**

When *number* is specified, if necessary, the response axis data range is extended to include the baseline intercept. When a logarithmic response axis is requested and *number* is 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINEINTERCEPT= to a positive value.

**AUTO**

bases the intercept on the response axis range.

The intercept is determined as follows:

- If the response axis range has both positive and negative values or contains 0, then the intercept is 0.
- If the response axis range contains all positive values, then AUTO is interpreted as AXISMINEXTEND.
- If the response axis range contains only negative values, then AUTO is interpreted as AXISMAXEXTEND.

**AXISMIN**

places the baseline at the minimum value of the axis range.

**AXISMINEXTEND**

places the baseline at the start of the minimum offset in the wall.

This location corresponds to the bottom edge of the wall when there is no inner margin plot and the axis is not reversed. If there is an inner margin plot at the bottom, then the baseline is placed at the boundary of the inner margin and the minimum offset.

**AXISMAX**

places the baseline at the maximum value of the axis range.

**AXISMAXEXTEND**

places the baseline at the start of the maximum offset in the wall.

This location corresponds to the top edge of the wall when there is no inner margin plot, and the axis is not reversed. If there is an inner margin plot at the top, then the baseline is placed at the boundary of the inner margin and the maximum offset.

**Default**

**AUTO**
Interactions

When `GROUPDISPLAY=STACK` is in effect, this option is ignored, and the plot is drawn as if `BASELINEINTERCEPT=0`.

When `DISPLAY=` includes `FILL`, the fill extends to the baseline that is specified by the `BASELINEINTERCEPT=` option.

**BREAK=TRUE | FALSE**

breaks the line at missing values of the `RESPONSE` variable.

Default: `FALSE`

Note: If `BREAK=FALSE`, then missing values are skipped and a continuous line is drawn.

See: "boolean" on page 1479 for other Boolean values that you can use.

**COLORMODEL=color-ramp-style-element | (color-list)**

specifies a color ramp to use with the `COLORRESPONSE=` option.

- `color-ramp-style-element` specifies the name of a color-ramp style element. The style element should contain these style attributes:
  - **STARTCOLOR** specifies the color for the smallest data value of the `COLORRESPONSE=` column.
  - **NEUTRALCOLOR** specifies the color for the midpoint of the range of the `COLORRESPONSE=` column.
  - **ENDCOLOR** specifies the color for the highest data value of the `COLORRESPONSE=` column.

- `(color-list)` specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as `GraphData3:Color`, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement: The list of colors must be enclosed in parentheses.

See: "color" on page 1480

Default: The ThreeColorAltRamp style element.

Interaction: For this option to take effect, the `COLORRESPONSE=` option must also be specified.

**COLORRESPONSE=numeric-column | range-attr-var | expression**

specifies the column or range attribute variable to use to map the line, marker, and fill colors.

Note: This feature applies to SAS 9.4M3 and to later releases.
range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

"RANGEATTRVAR Statement"

Restriction
A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See
"RANGEATTRVAR Statement" on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Requirement
The COLORRESPONSE values should remain constant for each group value in a grouped plot and for the entire plot in an ungrouped plot. If the COLORRESPONSE column has multiple values for a single GROUP value or for a non-grouped plot, unexpected results might occur.

Interactions
When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

When this option is specified without the GROUP= option, only a single line is generated for the plot, and the line color is derived from the COLORRESPONSE= value.

When fill is displayed, this option overrides suboption COLOR= in the FILLATTRS= option and varies the fill color according to the color gradient or the attribute map. The line and marker colors in that case are controlled by the contrastColor attribute of the GraphDataDefault style element.

When fill is not displayed, this option overrides suboption COLOR= in the LINEATTRS= and MARKERATTRS= options, and varies the line and marker colors according to the color gradient or the attribute map.

Tips
To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

For a numeric column or expression, the ThreeColorRamp style element defines the fill color gradient, and the ThreeColorAltRamp style element defines the line color gradient.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN enhances the visual appearance of the lines and markers. The following figure shows lines and CIRCLEFILLED markers with each of the skins applied.
The `DATASKIN=` option value that is specified in the `BEGINGRAPH` statement. If not specified, then the GraphSkins:DataSkin style element value is used.

Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the `DATASKINMAX=` option in your ODS GRAPHICS statement to increase the maximum limit.

This option overrides the `BEGINGRAPH` statement `DATASKIN=` option.

The skin appearance is based on the color that is in effect for the lines and markers.

**DATATRANSparency=** *number*

specifies the degree of the transparency of the line, markers, and vertex labels, if displayed.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

**Tip**

The `FILLATTRS=` option can be used to set transparency for just the fills. You can combine this option with `FILLATTRS=` to set one transparency for the lines but a different transparency for the fills. For example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**DISPLAY=** *(display-options)*  |  **STANDARD**  |  **ALL**

specifies which line-chart features to display.

`display-options`

a space-separated list of one or more display options enclosed in parentheses.

The list can include one or more of the following options:

**FILL**

displays a filled area between the line and the baseline
Interaction

When \texttt{GROUPDISPLAY=} STACK, \texttt{DISPLAY=} FILL fills between adjacent group lines except for the first group, which fills to the baseline.

LINE

displays line segments that join the vertices

MARKERS

displays markers at each vertex

\textbf{STANDARD}

specifies LINE only

\textbf{ALL}

specifies all features: FILL, LINE, and MARKERS

Default \textbf{STANDARD}

Tip

Use the \texttt{LINEATTRS=}, \texttt{MARKERATTRS=}, and \texttt{FILLATTRS=} options to control the appearance of the line segments, markers, and fill, respectively.

\textbf{FILLATTRS=} \texttt{style-element} | \texttt{(fill-options)}

specifies the appearance of the filled area.

Defaults

For non-grouped data, the \texttt{COLOR} attribute of GraphDataDefault style element.

For grouped data, the \texttt{COLOR} attribute of GraphData1–GraphDataN style elements is used.

Interactions

For this option to have any effect, the fill must be enabled by the ODS style or the \texttt{DISPLAY=} option.

When \texttt{COLORRESPONSE=} is in effect and the \texttt{DISPLAY=} option enables FILL display, the FILLATTRS= suboption \texttt{COLOR=} is ignored, and the fill colors vary according to the gradient.

Note

When \texttt{style-element} is specified, only the style element’s \texttt{COLOR} attribute is used.

Tip

The \texttt{DATATRANSPARENCY=} option sets the transparency for the fills and the lines. You can combine this option with \texttt{DATATRANSPARENCY=} to set one transparency for the lines but a different transparency for the fills. For example:

\texttt{datatransparency=0.2 fillattrs=(transparency=0.6)}

See

“General Syntax for Attribute Options” on page 1519

“Fill Color Options” on page 1520

\textbf{FILLEDOUTLINEDMARKERS=} \texttt{TRUE} | \texttt{FALSE}

specifies whether markers are drawn with both fill and an outline.

\texttt{TRUE}

draws filled markers (marker symbols with the suffix FILLED) using both fill and an outline.

When this option is \texttt{TRUE}, the fill color and outline color for filled markers are determined in the following ways:
If the GROUP= option is specified, then by default, the fill color is derived from the GraphData1–GraphDataN style elements Color attribute, and the marker outlined color is derived from the GraphData1–GraphDataN style elements ContrastColor attribute.

If the GROUP= option is not specified, then by default, the fill color is derived from the GraphDataDefault style elements Color attribute, and the marker outlined color is derived from the GraphOutline style elements ContrastColor attribute.

**FALSE**

draws the markers using fill or an outline, but not both.

**Default**  FALSE

**Tip**  To specify the marker fill and outline colors for a non-grouped plot, set this option to TRUE, and then use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify the colors.

**See**  GROUP= on page 591

MARKERFILLATTRS= on page 595

MARKEROUTLINEATTRS= on page 595

“boolean ” on page 1479 for other Boolean values that you can use.

**GROUP=column | discrete-attr-var | expression**

creates a separate line for each unique group value in the specified column.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction**  A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

**Default**  Each distinct group value is represented in the plot by a different combination of color, line pattern, and marker symbol. Lines and markers vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1–GraphDataN and GraphMissing style elements.

**Interaction**  When both the GROUP= and COLORRESPONSE= options are specified, the color attributes are controlled by the COLORRESPONSE= option.

**Tip**  The representations that are used to identify the groups can be overridden individually. For example, each distinct group value is represented by a different line pattern for the lines, but you can use the PATTERN= suboption of the LINEATTRS= option to assign the same line pattern to all lines.

**See**  “DISCRETEATTRMAP Statement” on page 1425
GROUPDISPLAY=OVERLAY | STACK
specifies how to display grouped lines.

OVERLAY
displays group values overlaid on top of each other.

STACK
displays group values as stacked lines.

Default: OVERLAY

Restriction: When STACK is in effect, if any response value is negative or if any crossing of the group value with the category is absent or is a missing value, then the chart is not drawn and a warning message is written to the SAS log.

Interaction: When STACK is in effect, the BASELINEINTERCEPT= option is treated as if it is set to zero.

Tip: When the response axis is linear, STAT=MEAN or STAT=PCT, and GROUPDISPLAY=STACK, the axis tick values might be displayed as integer values. When the response axis is linear, STAT=MEAN or STAT=PCT, and GROUPDISPLAY=OVERLAY, the axis tick values might be displayed as decimal values. To keep the integer axis values for both cases, you can specify the INTEGER=TRUE option for the response axis. See INTEGER= on page 1015.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in the group-column data order.

REVERSEDATA
orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip: This option is useful when you want to reverse the category axis.

ASCENDING
orders the groups within a category in ascending order.

DESCENDING
orders the groups within a category in descending order.

Defaults: DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

Interactions: This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes: When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When
plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

**Default** TRUE

**Interaction** For this option to take effect, the GROUP= option must also be specified.

**Tip** The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See** “boolean " on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**
specifies indices for mapping line attributes (color, marker symbol, and line pattern) and fill attributes to one of the GraphData1–GraphDataN style elements.

**Requirements**

- The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

- The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

- All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction** For this option to take effect, the GROUP= option must also be specified.

**Notes** The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.
If you do not use this option, then the group values are mapped in the order of the data.

Tip You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL= ”string”**

specifies a label to be used in a discrete legend for this plot.

Default If the **RESPONSE=** option is specified, then the response variable label is used. Otherwise, the **CATEGORY=** variable label is used. If a label is not assigned to the response variable or category variable, then the variable name is used.

Restriction This option applies only to an associated **DISCRETELEGEND** statement.

Interaction If the **GROUP=** option is specified, then this option is ignored.

**LINEATTRS= style-element | (line-options)**

specifies the appearance of the line.

Defaults For non-grouped data, the **GraphDataDefault** style element.

For grouped data, the **LineThickness** attributes of the **GraphDataDefault** style element, and the **ContrastColor** and **LineStyle** attributes of the **GraphData1–GraphDataN** style elements.

Interaction When **COLORRESPONSE=** is in effect, the **LINEATTRS=** suboption **COLOR=** is ignored, and the line fill colors vary according to the gradient.

Note When **style-element** is specified, only the style element’s **CONTRASTCOLOR**, **LINESTYLE**, and **LINETHICKNESS** attributes are used.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a **style-element**.

“Line Options” on page 1522 for available **line-options**.

**MARKERATTRS= style-element | style-element (marker-options) | (marker-options)**

specifies the attributes of the data markers.

Defaults For non-grouped data, **GraphDataDefault** style element.

For grouped data, the **MarkerSymbol** and **ContrastColor** attributes of the **GraphData1–GraphDataN** style elements, and the **GraphDataDefault:MarkerSize** style reference.

Interactions If **FILLEDOUTLINEDMARKERS=TRUE**, then this option’s **COLOR=** suboption is ignored. In that case, to specify the marker fill color, use the **MARKERFILLATTRS=** option instead.

This option’s **COLOR=** suboption overrides the default behavior for grouped data. When the **COLOR=** suboption is specified in that
case, all markers have the same color, and the marker symbol alone distinguishes the markers.

This option’s SYMBOL= suboption overrides the default behavior for grouped data. When the SYMBOL= suboption is specified in that case, all markers have the same symbol, and the symbol color alone distinguishes the markers.

The TRANSPARENCY= fill option overrides this option’s DATATRANSPARENCY= suboption.

This option is ignored if the DISPLAY= option disables the display of the markers.

If the DATASKIN= option is in effect, then the data skin determines the marker outlines. Any outline-related settings from the current ODS style or from the marker attribute options are ignored.

Note When style-element is specified, only the style element’s MARKERSYMBOL, CONTRASTCOLOR, and MARKERSIZE attributes are used.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Marker Options” on page 1523 for available marker-options.

MARKERFILLATTRS=style-element | (fill-options)
specifies the appearance of the filled markers.

Defaults For non-grouped data, the COLOR attribute of the GraphDataDefault style element

For grouped data, the COLOR attribute of a GraphData1–GraphDataN style element

Restriction The TRANSPARENCY= fill option is ignored. Use the MARKERATTRS= option to set the marker transparency.

Interaction This option is in effect only when FILLEDOUTLINEDMARKERS=TRUE and the DISPLAY= option enables fill display.

Note When style-element is specified, only the style element’s COLOR attribute is used.

See “General Syntax for Attribute Options” on page 1519

“Fill Color Options” on page 1520

MARKEROUTLINEATTRS=style-element | (line-options)
specifies the appearance of the marker outlines.

Defaults For non-grouped data, the GraphOutlines style element.

For grouped data, the LineThickness attribute of the GraphOutlines style element and the ContrastColor attribute of a GraphData1–GraphDataN style element.
| **Restriction** | The line style of the marker outline is always solid. |
| **Interaction** | This option is ignored when a data skin is applied by the current style or by the DATASKIN= option. In the latter case, the outline is set by the data skin. |
| **Note** | When *style-element* is specified, only the style element’s CONTRASTCOLOR and LINETHICKNESS attributes are used. |
| **See** | “General Syntax for Attribute Options” on page 1519 |
| | “Line Options” on page 1522 |

**NAME="string"**

assigns a name to this plot statement for reference in other template statements. This option is used mostly in the CONTINUOUSLEGEND statement in order to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**

The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**

The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**ORIENT=VERTICAL | HORIZONTAL**

specifies the orientation of the Y axis.

**Default** VERTICAL

**PRIMARY=TRUE | FALSE**

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default** FALSE

**Restriction**

This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note**

In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

**See**

“When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.
(role-name-list)
a space-separated list of role-name = column pairs.

Example  
The following example assigns the column Obs to the user-defined role TIP:  
ROLENAME=(TIP1=OBS)

Default  
No user-defined roles

Requirement  
The role names that you choose must be unique and different from the predefined roles CATEGORY, RESPONSE, DATALABEL, and GROUP.

Note  
The column values for a custom tip are not summarized using the STAT= statistic. For a given category value, the custom tip column values for that category value should be the same. Otherwise, unexpected results might occur.

SMOOTHCONNECT=TRUE | FALSE  
specifies that a smoothed line passes through all vertices.

Default  
FALSE. Straight line segments are used if the vertices are to be connected.

Interaction  
This option is ignored when GROUPDISPLAY=STACK.

See  
“boolean” on page 1479 for other Boolean values that you can use.

STAT=FREQ | PCT | PROPORTION | SUM | MEAN  
specifies the statistic to be computed for the Y axis. For line charts with no RESPONSE= column:

FREQ  
frequency count

PCT  
percentages between 0 and 100 inclusive

PROPORTION  
proportions between 0 and 1 inclusive

For line charts with a RESPONSE= column:

SUM

MEAN

Defaults  
SUM for line charts that specify the RESPONSE= argument.

FREQ for line charts that do not specify the RESPONSE= argument.

TIP=(role-list) | NONE  
specifies the information to display when the cursor is positioned over a line. If this option is used, then the information specified replaces all of the information that is displayed by default. You can specify roles for columns that do not contribute to the line chart along with roles that do.

(role-list)  
an ordered, space-separated list of unique LINECHART roles and user-defined roles.

LINECHART roles include CATEGORY, RESPONSE, COLORRESPONSE, DATALABEL, and GROUP.
The COLORRESPONSE role is valid starting with SAS 9.4M1.

User-defined roles are defined with the ROLENAME= option.

NONE suppresses data tips and URLs (if requested) from the plot.

The columns assigned to the following roles are automatically included in the data tip information: CATEGORY, RESPONSE, and GROUP.

Data tips are available only for graphs that are written to the ODS HTML destination.

To generate data tips, include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option. See "ODS GRAPHICS Statement" on page 1483 for information about the ODS GRAPHICS statement.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

The RESPONSE role represents the computed values for the CATEGORY role (and RESPONSE= role), based on the STAT= option.

You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

To display data tips for the columns assigned to the roles X and Y as well as the user-defined role TIP1:

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
```

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

```
(role-format-list)
```

a space-separated list of role-name = format pairs.

```
ROLENAME=(TIP1=salary)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
```

The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Only the roles that appear in the TIP= option are used.

A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

```
(role-label-list)
```

Example

```
ROLENAME=(TIP1=Salary)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
```
**role-label-list**

a space-separated list of `rolename = "string"` pairs.

**Example**

```
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

**Default**
The column label or column name of the column assigned to the role.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**URL=string-column**
specifies an HTML page to display when a line segment, marker, or fill area is selected.

**string-column**
specifies a column that contains a valid HTML page reference (HREF) for each line that is to have an active link.

**Example**

```
http://www.sas.com
```

**Requirement**
To generate selectable line segments, markers, and fill areas, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interactions**
This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Notes**
For non-grouped data, the values of the column are expected to be the same for each unique X value. If they are not, then the results might be unpredictable.

For grouped data, the values of the column are expected to be the same for each unique X and GROUP combination.

**Tips**
The URL value can be blank for some X values, meaning that no action is taken when the line segments for those X values are selected.

The URL value can be the same for different X values, meaning that the same action is taken when the line segments for those X values are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.
**VERTEXLABEL=TRUE | FALSE**

specifies whether to label the vertices with their response value (or statistic).

<table>
<thead>
<tr>
<th>Default</th>
<th>FALSE</th>
</tr>
</thead>
</table>

**Interaction**

When `GROUPDISPLAY=STACK` is in effect, vertex labeling displays the sum of the vertex responses per category.

**Note**

By default, the vertex-label format is derived from the format that is assigned to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

**Tips**

You can modify the visual attributes for the label by using the VERTEXLABELATTRS= option.

You can modify the text format by using the VERTEXLABELFORMAT= option.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**VERTEXLABELATTRS=style-element | style-element (text-options) | (text-options)**

specifies the color and font attributes of the vertex labels.

<table>
<thead>
<tr>
<th>Default</th>
<th>The GraphDataText style element.</th>
</tr>
</thead>
</table>

**Requirement**

`VERTEXLABEL=TRUE` must be in effect for this option to have any effect.

**Interaction**

If one or more text options are specified and they do not include all the font properties (color, family, size, weight, style), then non-specified properties are derived from the GraphDataText style element.

**Note**

When `style-element` is specified, only the style element's COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT attributes are used.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Text Options” on page 1525 for available `text-options`.

**VERTEXLABELFORMAT=format**

specifies the format used to display the vertex label response or statistic.

<table>
<thead>
<tr>
<th>Default</th>
<th>The vertex label format is derived from the format that is applied to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.</th>
</tr>
</thead>
</table>

**Requirement**

`VERTEXLABEL=TRUE` must be in effect for this option to have any effect.

**Note**

When a vertex-label format is specified with this option, the vertex labels are formatted as specified by `format`. The specified format is
Tip

If you want the vertex-label format to expand automatically for the response-column values, specify the format for the response column rather than in this option.

\textbf{XAXIS=}X \mid X2

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default \(X\)

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

\textbf{YAXIS=}Y \mid Y2

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default \(Y\)

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Optional Response Argument

\textbf{RESPONSE=}numeric-column | expression

specifies the numeric column or expression for the response values.

Details

A line chart shows the relationship of one variable to another as trends in the data over a period of time. The trends are shown by connecting the successive data points with a line. Typically, a line chart is used to chart a response value against a discrete categorical value where each value on the horizontal axis has only one corresponding value on the vertical axis. A grouping variable can be used to show multiple trends based on the group values.

In a line chart, the category axis (X) is always discrete, and the response axis (Y) is always linear. The line segments in the chart always join the categorical values in the order in which they appear on the axis. The vertices in the line chart appear in the order in which the categorical values appear in the input data.

Examples:

Example 1: Grouped Line Chart with Custom Line and Fill Attributes

This example shows you how to generate a simple line chart by using the LINECHART statement and how to use the statement options to customize the plot.

The following graph was generated by the “Example Program” on page 602. It shows the trend of the average closing price of the IBM, Intel, and Microsoft stocks from 1995 through 2005.
Example Program

Here is the SAS code for this example.

```sas
/* Extract per-year data for 1995 through 2005 from SASHELP.STOCKS */
data linechartdata;
    set sashelp.stocks(where=(year(date) between 1995 and 2005));
    year=year(date);
    label year="Year";
run;

/* Define the line chart template */
proc template;
    define statgraph linechart;
    begingraph;
        entrytitle "Stock Index Performance: 1995 - 2005";
        layout overlay /
            /* Add a grid */
            xaxisopts=(griddisplay=on gridattrs=(pattern=dot color=lightgray))
            yaxisopts=(griddisplay=on gridattrs=(pattern=dot color=lightgray));

        /* Generate the line chart */
        linechart category=year response=close / name="linechart"
            /* Compute the mean statistic */
            stat=mean
            /* Group by stock to draw a line for each stock */
            group=stock
    endgraph;
end;
```
Example 2: Grouped Line Chart with Discrete Attribute Map

This example shows you how to create a more flexible template that enables you to easily generate multiple line graphs that use different data. It also shows you how to customize the plot appearance by using a discrete attribute map.

The following graph was generated by the “Example Program” on page 604. It shows the trend of the average monthly closing price of the IBM, Intel, and Microsoft stocks for 2001.
Example Program

Here is the SAS code for this example.

```sas
/* Define the line chart template */
proc template;
    define statgraph linechart;
        begingraph;
            /* Create a dynamic variable for the year */
            dynamic year;

            /* Define the display attributes for each stock. Since DISCRETEATTRMAP does not support the MARKERATTRS= suboption SIZE=, the marker size is set separately. */
            discreteattrmap name="stocks" / ignorecase=true ;
                value "IBM" /
                    markerattrs=(color=blue symbol=trianglefilled)
                    lineattrs=(color=lightblue pattern=solid);
                value "Intel" /
                    markerattrs=(color=red symbol=circlefilled)
                    lineattrs=(color=verylightred pattern=solid);
                value "Microsoft" /
                    markerattrs=(color=orange symbol=squarefilled)
                    lineattrs=(color=verylightorange pattern=solid);
            enddiscreteattrmap ;

            /* Associate the attribute map with input data column Stock and assign the name STOCKATTRS to the named association */
            discreteattrvar attrvar=stockattrs var=stock attrmap="stocks" ;

            entrytitle "Stock Index Performance in " year;
            layout overlay /
                /* Add a grid */
                xaxisopts=(griddisplay=on gridattrs=(pattern=dot color=lightgray))
                yaxisopts=(griddisplay=on gridattrs=(pattern=dot color=lightgray));
            linechart category=date response=close / name="linechart"
                /* Compute the mean */
                stat=mean
                /* Group by stock using the specified attributes */
                group=stockattrs
            display=(line markers)
                /* Set the marker size */
                markerattrs=(size=6)
            /* Show vertex labels and specify their attributes */
                vertexlabel=true
                vertexlabelattrs=(size=7pt)
                vertexlabelformat=dollar4.0;

            /* Add a legend */
```
/* Create a macro that generates a line chart for a specific year */
%macro genchart(year=);
/* Generate the chart */
proc sgrender data=sashelp.stocks template=linechart;
  where year(date)=&year; /* Get the data for the specified year */
  format date monname3.; /* Format the date as 3-letter month */
  dynamic year="&year;"; /* Pass the year to the template */
run;
%mend genchart;

/* Generate a chart for 2001 */
%genchart(year=2001);

LINEPARM Statement

Creates a straight line specified by a point and a slope.

Requirement: A LINEPARM statement can be used only within a 2-D layout (OVERLAY, OVERLAYEQUATED, DATALATTICE, or DATAPANEL). Another plot statement that is derived from data values that provide boundaries for the axis area must be included.

Syntax

LINEPARM X=number | numeric-column | expression
Y=number | numeric-column | expression
SLOPE=number | numeric-column | expression <options>;

Summary of Optional Arguments

Appearance options

   CLIP=TRUE | FALSE
       specifies whether the data for the line is considered when the data ranges are determined for the axes.

   DATATRANSPARENCY=number
       specifies the degree of the transparency of the line.

   EXTEND=TRUE | FALSE
       specifies whether the line is to be drawn to the area bounded by the axes.

   INDEX=positive-integer-column | expression
       specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

   LINEATTRS=style-element | style-element (line-options) | (line-options)
       specifies the attributes of the line.

Axes options
XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Label options
CURVELABEL="string" | column | expression
specifies a label for the line.

CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the line label.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the line label relative to the plot area.

CURVELABELPOSITION=AUTO | MAX | MIN
specifies the position of the line label relative to the line end points.

CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the line label at the specified split characters.

CURVELABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the line label text.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the line label block.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options
GROUP=column | discrete-attr-var | expression
creates a separate parameterized line plot for each unique group value of the specified column.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments
X=number | numeric-column | expression
specifies the X coordinate of a point. By default, if the specified value is outside of the data range, then the data range is extended to include the specified intercept. This behavior can be changed with the CLIP= option. If a numeric-column is specified and the column contains missing values, then no line is drawn for the missing values.

Values are displayed in the units of the data.
**Y=number | numeric-column | expression**
specifies the Y coordinate of a point. By default, if the specified value is outside of the data range, then the data range is extended to include the specified intercept. This behavior can be changed with the **CLIP=** option. If a **numeric-column** is specified and the column contains missing values, then no line is drawn for the missing values.

Values are displayed in the units of the data.

**SLOPE=number | numeric-column | expression**
specifies the slope of the line. Slope can be positive or negative. **SLOPE=0** creates a line parallel to the X axis. **SLOPE=.** (a missing value) creates a line parallel to the Y axis.

**Optional Arguments**

**CLIP=TRUE | FALSE**
specifies whether the data for the line is considered when the data ranges are determined for the axes.

**FALSE**
specifies that the data for the line contributes to the data range for each axis.

Each axis might be extended in order to force the display of the line. When **CLIP=FALSE**, the **SLOPE=** option determines how the X= and Y= values contribute to the axis data range in the following ways:

- If **SLOPE=0**, then only the **Y=** values contribute to the axis data range.
- If **SLOPE=.** (missing), then only the **X=** values contribute to the axis data range.
- If **SLOPE=** is neither 0 nor missing, then the **X=** and **Y=** values contribute to the axis data range.

**TRUE**
specifies that the data for the line is ignored when axis scales are being established.

Each axis scale is determined by the other plots in the layout. In this case, the line might not be displayed if its data range is not within the data ranges of the other plots.

**Default**  **FALSE**

**See**  
“**boolean**” on page 1479 for other Boolean values that you can use.

**CURVELABEL="string" | column | expression**
specifies a label for the line.

**Default**  **No line label is displayed**

**Restrictions**
- When the GROUP= option is specified, "string" and expression are not valid. Use *column* in that case.
- When the GROUP= option is not specified, column is not valid. Use "string" or expression in that case.

The line label for missing values is ignored.
Tip
The font and color attributes for the label are specified by the CURVELABELATTRS= option.

See
GROUP= on page 612

CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the line label.

Defaults
For non-grouped data, the GraphValueText style element.

For grouped data, text color is derived from the GraphData1:ContrastColor–GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

Interactions
For this option to take effect, the CURVELABEL= option must also be used.

This option’s COLOR= setting overrides the colors indicated by the GROUP= option.

Tip
When the GROUP= option is used, each distinct group value might be represented by a different color. The line label that is associated with the group is assigned the group color. This option can be used to specify a single color for all line labels in a plot, without affecting the line colors.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the line label relative to the plot area.

INSIDE
locates the labels inside the plot area

OUTSIDE
locates the labels outside the plot area

Default
INSIDE

Restriction
OUTSIDE cannot be used when the LINEPARN is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interactions
For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELPOSITION= option to determine where the line labels appear.

See
“Location and Position of Curve Labels” on page 200
CURVELABELPOSITION=AUTO | MAX | MIN

specifies the position of the line label relative to the line end points. This option is used in conjunction with the CURVELABELLOCATION= option to determine where the line label appears.

AUTO

automatically positions the line label near the line boundary along unused axes whenever possible (typically Y2 and X2) in order to avoid collision with tick values.

Restriction  This option is used only when CURVELABELLOCATION= OUTSIDE.

MAX

forces the line label to appear near maximum line values (typically, upper right).

MIN

forces the line label to appear near minimum line values (typically, lower left).

Defaults  AUTO when CURVELABELLOCATION=OUTSIDE.

MAX when CURVELABELLOCATION=INSIDE.

Restriction  The AUTO setting is ignored if CURVELABELLOCATION= INSIDE is specified.

Interaction  For this option to take effect, the CURVELABEL= option must also be specified.

Note  When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the curve label might fall outside of the graph area. In that case, the curve label might not be displayed or might be positioned incorrectly.

See  “Location and Position of Curve Labels” on page 200

CURVELABELSPLIT=TRUE | FALSE

specifies whether to split the line label at the specified split characters. When a line label is split, the label is split on each occurrence of the specified split characters.

Default  FALSE. The line label is not split.

Requirement  The CURVELABEL= option must also be specified.

Interactions  The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.

This option has no effect when CURVELABELPOSITION=AUTO.

See  “boolean “ on page 1479 for other Boolean values that you can use.

CURVELABELSPLITCHAR="character-list"

specifies one or more characters on which the line label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear
consecutively in the line label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the line label is split unconditionally at each occurrence of any of the specified split characters. If the line label does not contain any of the specified characters, then the label is not split.

"character-list"

one or more characters with no delimiter between each character and enclosed in quotation marks.

<table>
<thead>
<tr>
<th>Default</th>
<th>A blank space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>The list of characters must be enclosed in quotation marks.</td>
</tr>
</tbody>
</table>

Multiple characters must be specified with no delimiters. For example, to specify the split characters a, b, and c, use the following option:

```
curvelabelsplitchar="abc"
```

The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

<table>
<thead>
<tr>
<th>Interactions</th>
<th>This option has no effect if CURVELABELPOSITION=AUTO.</th>
</tr>
</thead>
</table>

The CURVELABELSPLITCHARDROP= option specifies whether the split characters are included in the line label text.

```
CURVELABELSPLITCHARDROP=TRUE | FALSE
```

specifies whether the split characters are included in the line label text.

**TRUE**
drops the split characters from the line label text.

**FALSE**
includes the split characters in the line label text.

When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a line label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
CURVELABELSPLITCHAR="*"

Note: The horizontal line to the left of the label represents the maximum end of the line for reference.

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default: TRUE. The split characters are dropped from the line label.

Requirement: The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction: The CURVELABELSPLITCHAR= option specifies the split characters.

See "boolean " on page 1479 for other Boolean values that you can use.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the line label block.

AUTO

justifies the labels based on the CURVELABELPOSITION= option.

CURVELABELPOSITION= Value Justification

MAX or END LEFT

MIN or START RIGHT

CENTER

LEFT

RIGHT

justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the line for reference.
In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

**Default**

**AUTO**

**Requirement**
The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction**
This option has no effect if CURVELABELPOSITION=AUTO.

**DATATRANSPARENCY=number**

specifies the degree of the transparency of the line.

**Default**

0

**Range**

0–1, where 0 is opaque and 1 is entirely transparent

**Note**
This option does not affect the line label.

**EXTEND=TRUE | FALSE**

specifies whether the line is to be drawn to the area bounded by the axes.

**Default**

FALSE

**Note**
If this option is not specified, then there can be a small gap between the line and the axis. The gap is controlled by the axis offset. If the offset is set to 0, then there is no gap.

**See**
“boolean” on page 1479 for other Boolean values that you can use.

**GROUP=column | discrete-attr-var | expression**

creates a separate parameterized line plot for each unique group value of the specified column.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction**
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

**Default**
Each distinct group value might be represented in the plot by a different combination of line color and line pattern. Line colors vary according to the ContrastColor attribute of the GraphData1–GraphDataN and GraphMissing style elements. The line patterns vary according to the LineStyle attribute of the GraphData1–GraphDataN and GraphMissing style elements.
### Restriction
When this option is used, the X, Y, and SLOPE arguments must specify numeric columns.

### Interactions
The group values are mapped in the order of the data, unless the `INDEX=` option is used to alter the default sequence of line colors and line patterns.

The `INCLUDEMISSINGGROUP` option controls whether missing group values are considered a distinct group value.

### Note
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

### Tip
The `LINEATTRS=` option can be used to override the representations that are used to identify the groups. For example, `LINEATTRS=(PATTERN=SOLID)` can be used to assign the same pattern to all of the lines, letting the line color distinguish group values. Likewise, `LINEATTRS=(COLOR=BLACK)` can be used to assign the same color to all of the lines, letting the line pattern distinguish group values.

### See
"DISCRETEATTRVAR Statement" on page 1435

---

**INCLUDEMISSINGGROUP**=**TRUE** | **FALSE**
specifies whether missing values of the group variable are included in the plot.

**Default**
**TRUE**

**Interaction**
For this option to take effect, the `GROUP=` option must also be specified.

**Tip**
The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the `INDEX=` option is used, the `MISSING=` system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**
"boolean " on page 1479 for other Boolean values that you can use.

**INDEX**=**positive-integer-column** | **expression**
specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements**
The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.
Interaction
For this option to take effect, the GROUP= option must also be specified.

Notes
The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

Tip
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Default
The string specified on the NAME= option.

Restriction
This option applies only to an associated DISCRETELEGEND statement.

Interaction
If the GROUP= option is specified, then this option is ignored.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the line.

Defaults
For non-grouped data, the GraphDataDefault style element. For grouped data, the ContrastColor,LineStyle, and LineThickness attributes of the GraphData1–GraphDataN style elements.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element. “Line Options” on page 1522 for available line-options.

NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions
The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

Interaction
The string is used as the default legend label if the LEGENDLABEL= option is not used.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default
X
Restriction Another plot that establishes a data range for the designed axis must be included.

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

<table>
<thead>
<tr>
<th>Default</th>
<th>Y</th>
</tr>
</thead>
</table>

Restriction Another plot that establishes a data range for the designed axis must be included.

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details

The LINEPARM statement creates a straight line. You can generate a single line by specifying a constant for each required argument. You can generate multiple lines by specifying a numeric column for any or all required arguments. If any of the X= or Y= columns contains a missing value, then no line is drawn. To request a vertical line, specify `SLOPE=`. (specify a missing value as a constant or column value).

A LINEPARM statement can be used in any layout except GRIDDED or OVERLAY3D layouts. The parent layout must include another plot statement that is derived from data values that establish a data range for the axes. For example, it can be used with a scatter plot or a histogram.

To draw vertical or horizontal reference lines, consider using the simpler REFERENCeline statement.

Example: LINEPARM Statement

The following graph was generated by the “Example Program” on page 616:
Example Program

The LINEPARM statement draws a line based on a point and the slope of the line that passes through that point. You can use this statement to create a reference line with any slope or, in this example, to draw a fit from a linear regression. Many SAS/STAT procedures create output data sets containing a Y-intercept and slope and coefficient for the linear regression equation.

```sas
proc template;
define statgraph lineparm;
begingraph;
  entrytitle "Robust Fit of Height and Weight by Sex ";
  layout overlay / xaxisopts=(offsetmax=0.35);
  scatterplot x=height y=weight / group=sex
    markercharacter=eval(substr(sex,1,1))
    markercharacterattrs=(size=5pt) datatransparency=0.7;
  lineparm x=0 y=intercept slope=slope /
    name="Line" group=sex clip=true
    curvelabel=eval("Weight = ".||put(slope,5.3)||" * Height + ".||put(intercept,6.1))
    curvelabellocation=inside
    curvelabelattrs=(size=8pt);
  discretelegend "Line";
  endlayout;
endgraph;
end;
run;

proc sort data=sashelp.heart(keep=height weight sex)
   out=heart;
   by sex;
```
run;
ods exclude all;
proc robustreg data=heart method=m
   outest=stats(rename=(height=slope));
   by sex;
   model weight=height;
run;

data all;
   merge heart stats(keep=intercept slope sex);
run;
ods select all;
proc sgrender data=all template=lineparm;
run;

---

LOESSPLOT Statement

Creates a fitted loess curve computed from input data.

Restriction: The LOESSPLOT statement supports only models of one independent and one dependent variable.

Note: If the input data contains a large number of observations, then it might take several minutes to generate the plot.

Tips: By default, the LOESSPLOT statement will process up to 5000 observations. If the input data exceeds 5000 observations, then the plot is not generated. In that case, you can use the ODS GRAPHICS statement LOESSMAXOBS= option to extend the limit. See "Details" on page 628.

Starting with SAS 9.4M3, subpixel rendering is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see "ODS GRAPHICS Statement" on page 1483.

Syntax

**LOESSPLOT** *X=numeric-column | expression*

*Y=numeric-column | expression*  

\(<\text{regression-options}> \text{ <options>}>\);

Summary of Optional Arguments

Appearance options

- **DATATRANSPARENCY=number**
  specifies the degree of the transparency of the loess curve.

- **INDEX=positive-integer-column | expression**
  specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

- **LINEATTRS=style-element | style-element (line-options) | (line-options)**
  specifies the attributes of the loess curve.
Axes options

- **PRIMARY=TRUE | FALSE**
  specifies that the data columns for this plot and the plot type be used for determining default axis features.

- **XAXIS=X | X2**
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

- **YAXIS=Y | Y2**
  specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

- **TIPFORMAT=(role-format-list)**
  specifies display formats for tip columns.

- **TIPLABEL=(role-label-list)**
  specifies display labels for tip columns.

Label options

- **CURVELABEL="string"**
  specifies a label for the loess curve.

- **CURVELABELATTRS=style-element | style-element (text-options) | (text-options)**
  specifies the color and font attributes of the curve labels.

- **CURVELABELLOCATION=INSIDE | OUTSIDE**
  specifies the location of the curve label relative to the plot area.

- **CURVELABELPOSITION=AUTO | MAX | MIN | START | END**
  specifies the position of the curve label relative to the curve line.

- **CURVELABELSPLIT=TRUE | FALSE**
  specifies whether to split the curve label at the specified split characters.

- **CURVELABELSPLITCHAR="character-list"**
  specifies one or more characters on which the curve label can be split if needed.

- **CURVELABELSPLITCHARDROP=TRUE | FALSE**
  specifies whether the split characters are included in the curve label text.

- **CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**
  specifies the justification of the strings that are inside the curve label block.

- **LEGENDLABEL="string"**
  specifies a label to be used in a discrete legend for this plot.

Midpoint options

- **GROUP=column | discrete-attr-var | expression**
  creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

- **INCLUDEMISSINGGROUP=TRUE | FALSE**
  specifies whether missing values of the group variable are included in the plot.

Plot reference options

- **NAME="string"**
  assigns a name to this plot statement for reference in other template statements.
Required Arguments

\( X=\text{numeric-column | expression} \)

specifies the column for the X values.

\( Y=\text{numeric-column | expression} \)

specifies the column for the Y values.

Optional Arguments

CURVELABEL="string"

specifies a label for the loess curve.

Default

No curve label is displayed

Interaction

This option is not valid when the GROUP= option is specified.

Tip

The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the curve labels.

Default

The GraphValueText style element.

Interactions

For this option to take effect, the CURVELABEL= option must also be specified.

If the GROUP= option is specified, then this option is ignored.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the curve label relative to the plot area.

INSIDE

locates the labels inside the plot area

OUTSIDE

locates the labels outside the plot area

Default

INSIDE

Restriction

OUTSIDE cannot be used when the LOESSPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interactions

For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve labels appear.

See

“Location and Position of Curve Labels” on page 200
CURVELABELPOSITION=AUTO | MAX | MIN | START | END

specifies the position of the curve label relative to the curve line. This option is used in conjunction with the CURVELABELLOCATION= option to determine where the curve label appears.

**AUTO**
automatically positions the curve label near the curve boundary along unused axes whenever possible (typically Y2 and X2) in order to avoid collision with tick values.

*Restriction*  This option is used only when CURVELABELPOSITION=OUTSIDE.

**MAX**
forces the curve label to appear near maximum curve values (typically, upper right).

**MIN**
forces the curve label to appear near minimum curve values (typically, lower left).

**START**
forces the curve label to appear near the beginning of the curve.

*Restriction*  This option is used only when CURVELABELLOCATION=INSIDE.

*Tip*  This option is particularly useful when the curve line has a spiral shape.

**END**
forces the curve label to appear near the end of the curve.

*Restriction*  This option is used only when CURVELABELLOCATION=INSIDE.

*Tip*  This option is particularly useful when the curve line has a spiral shape.

**Defaults**
AUTO when CURVELABELLOCATION=OUTSIDE.

END when CURVELABELLOCATION=INSIDE.

**Interactions**
For this option to take effect, the CURVELABEL= option must also be specified.

The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

**Note**
When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the curve label might fall outside of the graph area. In that case, the curve label might not be displayed or might be positioned incorrectly.

*Tip*  When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.
CURVELABELSPLIT=TRUE | FALSE

specifies whether to split the curve label at the specified split characters. When a curve label is split, the label is split on each occurrence of the specified split characters.

Default

FALSE. The curve label is not split.

Requirement

The CURVELABEL= option must also be specified.

Interactions

The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.

This option has no effect when CURVELABELPOSITION=AUTO.

See

"boolean " on page 1479 for other Boolean values that you can use.

CURVELABELSPLITCHAR="character-list"

specifies one or more characters on which the curve label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the curve label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the curve label is split unconditionally at each occurrence of any of the specified split characters. If the curve label does not contain any of the specified characters, then the label is not split.

"character-list"

one or more characters with no delimiter between each character and enclosed in quotation marks.

Default

A blank space

Requirements

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no delimiters. For example, to specify the split characters a, b, and c, use the following option:

```
curvelabelsplitchar="abc"
```

The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interactions

This option has no effect if CURVELABELPOSITION=AUTO.

The CURVELABELSPLITCHARDROP= option specifies whether the split characters are included in the curve label or are dropped.

Notes

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.
Tip  Use the CURVELABELSPLITJUSTIFY= option to specify the justification of the strings in the curve label block.

**CURVELABELSPLITCHARDROP=TRUE | FALSE**  
specifies whether the split characters are included in the curve label text.  

**TRUE**  
drops the split characters from the curve label text.  

**FALSE**  
includes the split characters in the curve label text.  
When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.  

The following figure shows an example of a curve label with the following specifications:

- CURVELABELPOSITION=MAX  
- CURVELABEL="Product*Group*A"  
- CURVELABELSPLIT=TRUE  
- CURVELABELSPLITCHARDROP=TRUE | FALSE  
- CURVELABELSPLITCHAR="*"

**Note:** The horizontal line to the left of the label represents the maximum end of the curve for reference.

![Example of a curve label with split characters](image)

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.  

**Default**  
TRUE. The split characters are dropped from the curve label.  

**Requirement**  
The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.  

**Interaction**  
The CURVELABELSPLITCHAR= option specifies the split characters.  

**See**  
"boolean " on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**  
specifies the justification of the strings that are inside the curve label block.
AUTO
justifies the labels based on the CURVELABELPOSITION= option.

<table>
<thead>
<tr>
<th>CURVELABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
<tr>
<td>CENTER</td>
<td></td>
</tr>
<tr>
<td>LEFT</td>
<td></td>
</tr>
<tr>
<td>RIGHT</td>
<td></td>
</tr>
</tbody>
</table>

justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the curve for reference.

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

Default
AUTO

Requirement
The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction
This option has no effect if CURVELABELPOSITION=AUTO.

DATATRANSPARENCY=
number
specifies the degree of the transparency of the loess curve.

Default
0

Range
0–1, where 0 is opaque and 1 is entirely transparent

GROUP=
column | discrete-attr-var | expression
creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.
Each distinct group value might be represented in the plot by a different combination of color and line pattern. Line colors vary according to the ContrastColor attribute of the GraphData1–GraphDataN and GraphMissing style elements. Line patterns vary according to the LineStyle attribute of the GraphData1–GraphDataN style elements.

**Restriction**

The input data must be sorted by the GROUP= column.

**Interactions**

The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line colors and line patterns.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

**Tip**

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the loess curves, letting the line color distinguish group values. Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the curves, letting the line pattern distinguish group values.

**See**

“DISCRETEATTRVAR Statement” on page 1435

**INCLUDEMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the group variable are included in the plot.

**Default**

TRUE

**Interaction**

For this option to take effect, the GROUP= option must also be specified.

**Tip**

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**

“boolean ” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**

specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements**

The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.
### Interaction
For this option to take effect, the `GROUP=` option must also be specified.

### Notes
The index values are 1-based indices. For the style attributes in `GraphData1–GraphDataN`, if the index value is greater than `N`, then a modulo operation remaps that index value to a number less than `N` to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

### Tip
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 198.

### LEGENDLABEL=
`string`
specifies a label to be used in a discrete legend for this plot.

<table>
<thead>
<tr>
<th>Default</th>
<th>The <code>string</code> specified on the <code>NAME=</code> option.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>This option applies only to an associated <code>DISCRETELEGEND</code> statement.</td>
</tr>
<tr>
<td>Interaction</td>
<td>If the <code>GROUP=</code> option is specified, then this option is ignored.</td>
</tr>
</tbody>
</table>

### LINEATTRS=
`style-element | style-element (line-options) | (line-options)`
specifies the attributes of the loess curve.

<table>
<thead>
<tr>
<th>Default</th>
<th>The <code>GraphFit</code> style element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>See</td>
<td>&quot;General Syntax for Attribute Options&quot; on page 1519 for the syntax on using a <code>style-element</code>.</td>
</tr>
<tr>
<td></td>
<td>&quot;Line Options&quot; on page 1522 for available <code>line-options</code>.</td>
</tr>
</tbody>
</table>

### NAME=
`string`
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

<table>
<thead>
<tr>
<th>Restrictions</th>
<th>The <code>string</code> is case sensitive, cannot contain spaces, and must define a unique name within the template.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This option does not support variables that are created by the <code>DYNAMIC</code>, <code>MVAR</code>, and <code>NMVAR</code> template statements.</td>
</tr>
<tr>
<td>Interaction</td>
<td>The <code>string</code> is used as the default legend label if the <code>LEGENDLABEL=</code> option is not used.</td>
</tr>
</tbody>
</table>

### PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

<table>
<thead>
<tr>
<th>Default</th>
<th><code>FALSE</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>This option is ignored if the plot is placed under a <code>GRIDDED</code> or <code>LATTICE</code> layout block.</td>
</tr>
</tbody>
</table>
Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example TIPFORMAT=(Y=6.2)

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Requirement To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Note The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the data tip information.

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example TIPLABEL=(Y="Curve")

Default The column label or column name of the column assigned to the role.

Note The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the data tip information.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.
Loess Regression Options

**ALPHA=positive-number**
- specifies the confidence level to compute.

  Default: 0.05

  Range: $0 < \text{number} < 1$

  Tip: ALPHA=0.05 represents a 95% confidence level.

**CLM=“name”**
- produces confidence limits for a mean predicted value for each observation. The confidence level is set by the ALPHA= option.

  Interaction: "name" is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBand statement.

**DEGREE=1 | 2**
- specifies the degree of the local polynomials to use for each local regression. The valid values are 1 for local linear fitting or 2 for local quadratic fitting.

  Default: 1

**INTERPOLATION=LINEAR | CUBIC**
- specifies the degree of the interpolating polynomials used for blending local polynomial fits at the kd tree vertices.

  **CUBIC**
  - cubic polynomials

  **LINEAR**
  - linear polynomials

  Default: LINEAR

**MAXPOINTS=positive-integer**
- specifies the maximum number of predicted points generated for the loess curve as well as confidence limits.

  Default: 201

**SMOOTH=AUTO | positive-number**
- specifies a regression parameter value.

  Default: AUTO
REWEIGHT=NONE | positive-integer
specifies the number of iterative re-weighting steps to be done. Such iterations are appropriate when there are outliers in the data or when the error distribution is a symmetric long-tailed distribution.

Default NONE

WEIGHT=numeric-column
specifies a column in the input data set that contains values to be used as a priori weights for a loess fit. The values of the weight column must be nonnegative. If an observation’s weight is zero, negative, or missing, then the observation is deleted from the analysis.

Details
The LOESSPLOT statement supports only statistical models of one independent and one dependent variable. For more information about the fitting methodology, see the LOESS procedure in the SAS/STAT user’s guide.

In addition to the loess curve, the LOESSPLOT statement can compute confidence levels for the fitted line. To display the confidence levels:

1 Use the CLM= option to declare a name for the confidence level of the mean.
2 Use a MODELBAND statement to refer this name. This statement draws a confidence band from this information. See “MODELBAND Statement” on page 629 for information about how to control the appearance of the confidence band.

By default, the LOESSPLOT statement will process up to 5000 observations. If the input data contains more than 5000 observations, then the plot is not drawn and the following note is written to the SAS log:

NOTE: The number of observations of the LOESS plot (nnnn) exceeds the limit of 5000. Specify the LOESSMAXOBS option of the ODS GRAPHICS statement to override the limit.

In that case, you can use the following statement to extend the limit:

ods graphics / loessmaxobs=nnnn

where nnnn is the new limit.

Note: When the input data contains a large number of observations, it might take several minutes to generate the plot.

For more information about the LOESSMAXOBS= option, see “ODS GRAPHICS Statement” on page 1483.

Example: LOESSPLOT Statement
The following graph was generated by the “Example Program” on page 629:
Example Program

```sas
proc template;
define statgraph loessplot;
begingraph;
  entrytitle "Loess Fit Plot";
  layout overlay;
    scatterplot x=weight y=mpg_highway / datatransparency=0.7;
    loessplot x=weight y=mpg_highway / name="fitline" alpha=0.05 legendlabel="Loess Fit";
    discretelegend "fitline";
  endlayout;
endgraph;
end;
run;

proc sgrender data=sashelp.cars template=loessplot;
run;
```

MODELBAND Statement

Creates a band showing confidence limits for an associated smoother plot.

Requirement: A MODELBAND statement must be associated with a smoother statement (LOESSPLOT, REGRESSIONPLOT, or PBSPLINEPLOT) that specifies a fitted model and a type of confidence level to compute.
Starting with SAS 9.4M2, a confidence band that depicts confidence limits for individual predicted values (CLI) for a weighted spline plot or regression plot is displayed as a high-low chart instead of a band.

Syntax

MODEL_BAND "$confidence-name" <options>;

Summary of Optional Arguments

Appearance options

ANTIALIAS=AUTO | OFF
specifies whether anti-aliasing is turned off for this plot.

DATATRANSPARENCY=number
specifies the degree of the transparency of the band fill and the band outline.

DISPLAY=(display-options) | STANDARD | ALL
specifies which band features to display.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the filled modelband area.

FILLPATTERNATTRS=style-element | (fill-pattern-options)
specifies the appearance of the pattern-filled areas.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the modelband outlines.

Axes options

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

TIPFORMAT=(role-format-list)
specifies display formats for tip columns.

TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options

CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the upper and lower band labels.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the band labels relative to the plot area.

CURVELABELLOWER="string"
specifies a label for the lower band limit.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
specifies the position of the band label relative to the band line.

CURVELABELSSPLIT=TRUE | FALSE
specifies whether to split the curve labels at the specified split characters.

**CURVELABELSPLITCHAR=**"character-list"

specifies one or more characters on which the curve labels can be split if needed.

**CURVELABELSPLITCHARDROP=**TRUE | FALSE

specifies whether the split characters are included in the curve label text.

**CURVELABELSPLITJUSTIFY=**AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the curve label blocks.

**CURVELABELUPPER=**"string"

specifies a label for the upper band limit.

**LEGENDLABEL=**"string"

specifies a label to be used in a discrete legend for this plot.

**Plot reference options**

**NAME=**"string"

assigns a name to this plot statement for reference in other template statements.

**Required Argument**

**"confidence-name"**

specifies the case-sensitive name assigned to a confidence option in a smoother plot statement.

**Requirement**  
*confidence-name* must have been assigned to the CLM= or CLI= option on a smoother plot statement such as LOESSPLOT, REGRESSIONPLOT, or PBSPLINEPLOT.

**Optional Arguments**

**ANTIALIAS=**AUTO | OFF

specifies whether anti-aliasing is turned off for this plot.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**AUTO**

specifies that anti-aliasing is controlled by the ANTIALIAS= option in the ODS GRAPHICS statement.

**OFF**

specifies that anti-aliasing is always disabled for this plot.

**Default** AUTO

**Interaction**  
This option overrides the ANTIALIAS= option in the ODS GRAPHICS statement.

**CURVELABELATTRS=**style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the upper and lower band labels.

**Default**  
The GraphValueText style element.
Interactions

For this option to take effect, the `CURVELABELLOWER=` or `CURVELABELUPPER=` option must also be specified.

If the smoother statement’s GROUP= option is specified, then this option is ignored.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Text Options” on page 1525 for available `text-options`.

**CURVELABELLOWER=“string”**

specifies a label for the lower band limit.

**Default**

No curve label is displayed for the lower band

**Interaction**

If the smoother statement’s GROUP= option is specified, then this option is ignored.

**Tip**

The font and color attributes for the label are specified by the `CURVELABELATTRS=` option.

**CURVELABELUPPER=“string”**

specifies a label for the upper band limit.

**Default**

No curve label is displayed for the upper band

**Interaction**

If the smoother statement’s GROUP= option is specified, then this option is ignored.

**Tip**

The font and color attributes for the label are specified by the `CURVELABELATTRS=` option.

**CURVELABELLOCATION=INSIDE | OUTSIDE**

specifies the location of the band labels relative to the plot area.

**INSIDE**

locates the labels inside the plot area

**OUTSIDE**

locates the labels outside the plot area

**Default**

INSIDE

**Restriction**

OUTSIDE cannot be used when the MODELBand is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

**Interactions**

For this option to take effect, the `CURVELABELLOWER=` or `CURVELABELUPPER=` option must also be specified.

This option is used in conjunction with the `CURVELABELPOSITION=` option to determine where the band labels appear.

**See**

“Location and Position of Curve Labels” on page 200

**CURVELABELPOSITION=AUTO | MAX | MIN | START | END**

specifies the position of the band label relative to the band line.
**AUTO**
automatically positions the band labels near the band boundary along unused axes whenever possible (typically Y2 and X2).

**Restriction**  This option is used only when CURVELABELPOSITION=OUTSIDE.

**MAX**
forces the band label to appear near maximum band values (typically, upper right)

**MIN**
forces the band label to appear near minimum band values (typically, lower left)

**START**
forces the band label to appear near the beginning of the band.

**Restriction**  This option is used only when CURVELABELLOCATION=INSIDE.

**Tip**  This option is particularly useful when the curve line has a spiral shape.

**END**
forces the band label to appear near the end of the band.

**Restriction**  This option is used only when CURVELABELLOCATION=INSIDE.

**Tip**  This option is particularly useful when the curve line has a spiral shape.

**Defaults**
AUTO when CURVELABELLOCATION=OUTSIDE.
END when CURVELABELLOCATION=INSIDE.

**Restrictions**
The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified.
The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

**Interactions**
For this option to take effect, the CURVELABELLOWER= or CURVELABELUPPER= option must also be specified.
This option is used in conjunction with the CURVELABELLOCATION= option to determine where the band label appears.

**Note**
When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the band label might fall outside of the graph area. In that case, the band label might not be displayed or might be positioned incorrectly.

**Tip**  When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.
See "Location and Position of Curve Labels" on page 200

**CURVELABELSPLIT=TRUE | FALSE**

Specifies whether to split the curve labels at the specified split characters. When a curve label is split, the label is split on each occurrence of the specified split characters.

- **Default**: FALSE. The curve label is not split.
- **Requirement**: The CURVELABEL= option must also be specified.
- **Interactions**: The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.
- **See**: “boolean “ on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITCHAR=character-list**

Specifies one or more characters on which the curve labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the curve label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the curve labels are split unconditionally at each occurrence of any of the specified split characters. If the curve label does not contain any of the specified characters, then the label is not split.

- **character-list**: one or more characters with no delimiter between each character and enclosed in quotation marks.
- **Default**: A blank space
- **Requirements**: The list of characters must be enclosed in quotation marks.
- **Interactions**: This option has no effect if CURVELABELPOSITION=AUTO.
- **Notes**: When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.
Tip
Use the CURVELABELSPLITJUSTIFY= option to specify the justification of the strings in the curve label blocks.

**CURVELABELSPLITCHARDROP=TRUE | FALSE**
specifies whether the split characters are included in the curve label text.

**TRUE**
drops the split characters from the curve label text.

**FALSE**
includes the split characters in the curve label text.

When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a curve label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

Note: The horizontal line to the left of the label represents the maximum end of the series line for reference.

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**
TRUE. The split characters are dropped from the curve labels.

**Requirement**
The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction**
The CURVELABELSPLITCHAR= option specifies the split characters.

**See**
"boolean " on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**
specifies the justification of the strings that are inside the curve label blocks.
AUTO
justifies the labels based on the CURVELABELPOSITION= option.

<table>
<thead>
<tr>
<th>CURVELABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

CENTER
LEFT
RIGHT
justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the band for reference.

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

Default AUTO
Requirement The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.
Interaction This option has no effect if CURVELABELPOSITION=AUTO.

DATATRANSAPRENCY=number
specifies the degree of the transparency of the band fill and the band outline.

Default 0
Range 0–1, where 0 is opaque and 1 is entirely transparent
Note This option does not affect the curve label.
Tip The FILLATTRS= option can be used to set transparency for just the filled band area. You can combine this option with FILLATTRS= to set one transparency for the band outline but a different transparency for the band fill. Example:

datatransparency=0.2 fillattrs=(transparency=0.6)

DISPLAY=(display-options) | STANDARD | ALL
specifies which band features to display.
(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

- **FILL**
  - displays the band fill color

- **OUTLINE**
  - displays the band outline

- **STANDARD**
  - specifies FILL only

- **ALL**
  - specifies all features: FILL and OUTLINE

**Default**
The GraphBand:DisplayOpts style reference.

**Tip**
Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the band.

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the filled modelband area.

**Defaults**
For non-grouped data, the GraphConfidence:Color style reference.
For grouped data, the GraphData1:Color–GraphDataN:Color style references.

**Interaction**
For this option to have any effect, the fill must be enabled by the ODS style or the DISPLAY= option.

**Tip**
The DATATRANSPARENCY= option sets the transparency for band outline and the band fill. You can combine this option with DATATRANSPARENCY= to set one transparency for the band outline but a different transparency for the band fill. Example:

datatransparency=0.2 fillattrs=(transparency=0.6)

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

**FILLPATTERNATTRS=style-element | (fill-pattern-options)**
specifies the appearance of the pattern-filled areas.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Defaults**
For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.
For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

**Interaction**
For this option to take effect, the DISPLAY= option must include FILLPATTERN among the display options.
Tip
In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
“Fill Pattern Options” on page 1521 for available fill-pattern-options.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Default The string specified on the NAME= option.

Restriction This option applies only to an associated DISCRETELEGEND statement.

Interaction The smoother statement’s GROUP= option overrides this option.

NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction The string is used as the default legend label if the LEGENDLABEL= option is not used.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the modelband outlines.

Defaults For non-grouped data, the GraphConfidence style element.

For grouped data, the GraphData1: ContrastColor–GraphDataN:ContrastColor style references.

Interaction If DISPLAY=(FILL), then this option has no effect.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
“Line Options” on page 1522 for available line-options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs. The role-names X, LIMITLOWER, LIMITUPPER, GROUP, and INDEX are available to indicate which data tip values to format.
Example: TIPFORMAT=(LIMITUPPER=5.3 LIMITLOWER=5.3)

Default
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Requirement
To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename="string" pairs. The role-names X, LIMITLOWER, LIMITUPPER, GROUP, and INDEX are available to indicate which data tip values to label.

Example
TIP=(X)
TIPLABEL=(X="Type")

Default
The column label or column name of the column assigned to the role.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default
X

Requirement
The setting for this option should be the same as for the smoother statement referenced by the confidence-name.

Interaction
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default
Y

Requirement
The setting for this option should be the same as for the smoother statement referenced by the confidence-name.

Interaction
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Example: MODELBAND Statement
The following graph was generated by the “Example Program” on page 640:
Example Program

```sas
proc template;
define statgraph modelband;
begingraph;
    entrytitle "Spline Fit with Confidence Bands";
    layout overlay;
        modelband "cliband" / name="confband1" display=all legendlabel="90% CLI" fillattrs=GraphConfidence;
        modelband "clmband" / name="confband2" display=all legendlabel="90% CLM" fillattrs=GraphConfidence2;
        scatterplot x=weight y=mpg_highway /
            datatransparency=0.7;
        pbsplineplot x=weight y=mpg_highway / name="fitline"
            clm="clmband" cli="cliband"
            alpha=0.1 legendlabel="Spline Fit";
        discretelegend "fitline" "confband1" "confband2";
    endlayout;
endgraph;
end;
run;

proc sgrender data=sashelp.cars template=modelband;
run;
```

MOSAICPLOTPARM Statement

Creates a mosaic plot from pre-summarized categorical data.
Restriction: You can use the MOSAICPLOTPARM statement in GRIDDED, LATTICE, and REGION layouts only.

Note: On z/OS hosts, the mosaic plot categories might appear in an order other than data order.

Syntax

MOSAICPLOTPARM CATEGORY=(column-list)
COUNT=non-negative-numeric-column | expression </options>;

Summary of Optional Arguments

Appearance options

COLORGROUP=column | discrete-attr-var
specifies the category column to use for discrete fill colors for the tiles.

COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies a numeric column to use to map tile fill colors to a continuous gradient.

DATATRANSPARENCY=number
specifies the degree of the transparency of the tile fill, outlines, and the values that are located inside the tiles if those values are displayed.

DISPLAY=(display-options) | STANDARD | ALL
specifies which mosaic-plot features to display.

FILLATTRS=style-element | (fill-options)
specifies the appearance of the tile fill areas.

GUTTER=dimension | (dimension-list)
specifies the gutter (gap) between the splits.

OUTLINEATTRS=style-element | (line-options)
specifies the appearance of the tile outlines.

REVORSECOLORMODEL=TRUE | FALSE
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

SQUARED=TRUE | FALSE
specifies that a square aspect ratio be used for the plot area.

Data tip options

ROLENAME=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips.

TIP=(role-list) | NONE
specifies the information that is displayed when the cursor is positioned over a tile.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns.

TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Label options
INSIDEVALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the category values when they are located inside a tile.

LABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the category labels.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

VALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the category values that are located outside of the tiles.

VALUELOCATION=AUTO | INSIDE | OUTSIDE
specifies the location of the category column values in a two-way plot.

XVALUEFITPOLICY=ROTATE | NONE
specifies a policy for avoiding collisions along the width of the plot among category values that are outside of the tiles.

YVALUEFITPOLICY=NONE | ROTATEALWAYS
specifies a policy for avoiding collisions along the height of the plot among category values that are outside of the tiles.

ODS options
URL=string-column
specifies the URL of an HTML page to display when a tile is selected.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments

CATEGORY=(column-list)
specifies a list of columns of category (classification) values.

Restriction  No more than three columns can be specified.

Note  When a SAS data set is plotted, the category values are arranged in data order. When a CAS in-memory table is plotted, the category values are arranged in ascending order of the formatted character values or the unformatted numeric values of the category column or columns.

COUNT=non-negative-numeric-column | expression
specifies the column of counts (pre-summarized) for each of the category value combinations.

Restriction  The column values cannot be negative.

Tip  You need to provide only the category crossings with nonzero counts.

Optional Arguments

COLORGROUP=column | discrete-attr-var
specifies the category column to use for discrete fill colors for the tiles.
**discrete-attr-var**
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a **DISCRETEATTRVAR** statement.

**Restriction**
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Each unique value of this column is mapped to the COLOR attribute of the GraphData1–GraphDataN style elements that are in effect. If a discrete attribute map variable is specified, the color mapping from its associated **DISCRETEATTRMAP** statement is used.

**Restriction**
This column or the associated column in the discrete attribute map variable must be one of the columns in the category column list.

**Interactions**
This option is ignored if the **COLORRESPONSE=** option is specified.

This option overrides the **FILLATTRS=** option.

**Note**
If you specify a column in a SAS data set, colors are mapped to group values in data order. If you specify a column in a CAS in-memory table, colors are mapped to group values in ascending order of the group column character values or unformatted numeric values.

**COLORMODEL=** **color-ramp-style-element | (color-list)**
specifies a color ramp to use with the **COLORRESPONSE=** option.

**color-ramp-style-element**
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

**STARTCOLOR**
specifies the color for the smallest data value of the **COLORRESPONSE=** column.

**NEUTRALCOLOR**
specifies the color for the midpoint of the range of the **COLORRESPONSE=** column.

**ENDCOLOR**
specifies the color for the highest data value of the **COLORRESPONSE=** column.

**(color-list)**
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement**
The list of colors must be enclosed in parentheses.

**See**
“**color** ” on page 1480

**Default**
The ThreeColorRamp style element
For this option to take effect, the COLORRESPONSE= option must also be specified.

Tip

To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

**COLORRESPONSE=**`numeric-column | range-attr-var | expression`

designates a numeric column to use to map tile fill colors to a continuous gradient.

**range-attr-var**

specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

"RANGEATTRVAR Statement"

**Restriction**

A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

**See**

"RANGEATTRVAR Statement" on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

**Interactions**

When fill is displayed, this option overrides suboption COLOR= in the FILLATTRS= option and varies the fill color according to the color gradient or the attribute map.

When only the outlines are displayed, this option overrides suboption COLOR= in the OUTLINEATTRS= option and varies the outline color according to the color gradient or the attribute map.

**Tip**

To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

**DATATRANSPARENCY=**`number`

designates the degree of the transparency of the tile fill, outlines, and the values that are located inside the tiles if those values are displayed.

**Default**

0

**Range**

0–1, where 0 is opaque and 1 is entirely transparent

**DISPLAY=(display-options) | STANDARD | ALL**

designates which mosaic-plot features to display.

**display-options**

a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**FILL**

displays filled tiles

**OUTLINE**

displays the tile outline
LABELS
displays the category column labels

TICKS
displays the category ticks

VALUES
displays the category values

STANDARD
specifies FILL, OUTLINE, LABELS, and VALUES

ALL
specifies all features: FILL, OUTLINE, LABELS, TICKS, and VALUES

Default STANDARD

Interactions If neither FILL nor OUTLINE are present in the display-options list, then filled and outlined tiles are displayed.

If YVALUELOCATION=INSIDE or if YVALUELOCATION=AUTO and is effectively set to INSIDE, then the axis ticks are not displayed even if the display of the ticks is specified for the axis.

FILLATTRS=style-element | (fill-options)
specifies the appearance of the tile fill areas. See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element. See “Fill Color Options” on page 1520 for the available fill options.

Default The GraphDataDefault style element

Interaction When COLORRESPONSE= is in effect and the DISPLAY= option enables FILL display, the FILLATTRS= suboption COLOR= is ignored, and the fill colors vary according to the gradient.

Note When style-element is specified, only the style element’s COLOR attribute is used.

GUTTER=dimension | (dimension-list)
specifies the gutter (gap) between the splits. The splits occur in the following way:

- When a single dimension is specified, the dimension applies to the gap for the last split, which has the smallest gap. From the next-to-last last split to the first split, the gutter is doubled on each split. A single dimension has the effect of setting a minimum gap for the plot. The following figure shows an example in which GUTTER=10.
Note: The plot contains equally sized tiles for demonstration purposes.

In this case, the gaps are 30 pixels for the first split, 20 pixels for the second split, and 10 pixels for the third split (minimum gap).

- When a list of dimension values is used, the values apply to each split in the order in which they are specified. The following figure shows an example in which GUTTER=(10 20 30).

In this case, the gaps are 10 pixels for the first split, 20 pixels for the second split, and 30 pixels for the third split.

Default 3px. Dimensions with no units are assumed to be in pixels.

See “dimension” on page 1480

INSIDEVALUEATTRS=style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the category values when they are located inside a tile. See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element. See “Text Options” on page 1525 for the available text options.

Default The GraphValueText style element.

Interaction If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then
the non-specified properties are derived from the GraphValueText style element.

**Note**
When *style-element* is specified, only the element’s COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT attributes are used.

**Tip**
You can use the `VALUEATTRS=` option to change the text attributes for the values that are located outside the plot area.

<table>
<thead>
<tr>
<th><strong>LABELATTRS=</strong> *style-element</th>
<th><em>style-element</em> (<em>text-options</em>)</th>
<th>(<em>text-options</em>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies the color and font attributes of the category labels. See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element. See “Text Options” on page 1525 for the available text options.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Default**
The GraphLabelText style element.

**Interaction**
If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the non-specified properties are derived from the GraphLabelText style element.

**Note**
When *style-element* is specified, only the style element’s COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT attributes are used.

<table>
<thead>
<tr>
<th><strong>LEGENDLABEL=</strong> &quot;string&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies a label to be used in a discrete legend for this plot.</td>
</tr>
</tbody>
</table>

**Default**
The `string` specified on the `NAME=` option.

**Restriction**
This option applies only to an associated `DISCRETELEGEND` statement.

**Interaction**
If the `COLORGROUP=` option is in effect, then this option is ignored.

<table>
<thead>
<tr>
<th><strong>NAME=</strong> &quot;string&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.</td>
</tr>
</tbody>
</table>

**Restrictions**
The `string` is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**
The `string` is used as the default legend label if the `LEGENDLABEL=` option is not used.

<table>
<thead>
<tr>
<th><strong>OUTLINEATTRS=</strong> *style-element</th>
<th>(<em>line-options</em>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies the appearance of the tile outlines. See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element. See “Line Options” on page 1522 for the available line options.</td>
<td></td>
</tr>
</tbody>
</table>

**Default**
The GraphOutlines style element.
Interaction When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the outline colors vary according to the gradient.

Note When style-element is used, only the style element’s CONTRASTCOLOR, LIFESTYLE, and LINETHICKNESS are used.

**REVERSECOLORMODEL=TRUE | FALSE**

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

**Default** FALSE

**See** COLORMODEL=

“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAMESPACE=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

*(role-name-list)*

a space-separated list of role-name = column pairs.

**Example**
The following example assigns the column Obs to the user-defined role TIP:

ROLENAMESPACE=(TIP=OBS)

**Default** No user-defined roles

**Requirement**
The role names that you choose must be unique and different from the predefined roles. The predefined roles are CATEGORY1–CATEGORYn (in the order in which they are specified in the CATEGORY= option), COUNT, COLORGROUP, and COLORRESPONSE.

**SQUARED=TRUE | FALSE**

specifies that a square aspect ratio be used for the plot area.

**Default** FALSE

**Restriction** This option applies to multi-way plots only.

**Tip** Setting this option to TRUE makes the height of the plot the same as its width, which can make it easier to compare the proportions.

**See** “boolean ” on page 1479 for other Boolean values that you can use.

**TIP=(role-list) | NONE**

specifies the information that is displayed when the cursor is positioned over a tile. If this option is used, then all of the information that is displayed by default is replaced. Roles for columns that do not contribute to the bar chart can be specified along with roles that do.

*(role-list)*

an ordered, space-separated list of unique MOSAICPLOTPARM roles and user-defined roles.
The MOSAICPLOTPARM roles include COUNT, COLORGROUP, and COLORRESPONSE. User-defined roles are defined with the ROLENAMEx= option.

**NONE**  
suppresses data tips and URLs (if requested) from the plot.

**Default**  
The columns that are assigned to the category columns and COUNT roles are automatically included in the data tip information.

**Requirement**  
To generate data tips, you must include an ODS GRAPHICS ON statement with the IMAGEMAP option specified. You must also write the graphs to the ODS HTML destination.

**Interaction**  
The labels and formats for the TIP roles can be controlled with the TIPLABELx= and TIPFORMATx= options.

**Example**  
To display data tips for the columns that are assigned to the roles X and Y as well as the user-defined role TIP1:

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
```

**TIPLABELx=(role-label-list)**  
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

*role-label-list*  
a space-separated list of rolename “=string” pairs.

**Example**  
```
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

**Default**  
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**  
Only the roles that appear in the TIP= option are used.

**Requirement**  
A column must be assigned to each of the specified roles. (See the ROLENAMEx= option.)

**Note**  
This option is no longer supported starting with SAS 9.4M4. To format tip values in SAS 9.4M4 and in later releases, use a FORMAT statement in the SGRENDER procedure step. See “FORMAT Statement” in SAS ODS Graphics: Procedures Guide.

**TIPFORMATx=(role-format-list)**  
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

*role-format-list*  
a space-separated list of role-name “=format” pairs.

**Example**  
```
ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
```

**Default**  
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.
The column label or column name of the column assigned to the role.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**URL=string-column**
specifies the URL of an HTML page to display when a tile is selected.

*string-column*
specifies a column that contains a valid HTML page reference (HREF) for each tile that is to have an active link.

*Example* http://www.sas.com

Requirement
To generate selectable tiles, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option. You must also write the graphs to the ODS HTML destination.

Tips
The URL value can be blank for some tiles, which means that no action is taken when those tiles are selected. The URL value can be the same for different tiles, which means that the same action is taken when those tiles are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See **DRILLTARGET=** on page 1489.

**VALUEATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the category values that are located outside of the tiles. See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element. See “Text Options” on page 1525 for the available text options.

Default
The GraphValueText style element.

Interaction
If one or more text options are specified and they do not include all the font properties such as color, family, size, weight, and style, then the non-specified properties are derived from the GraphValueText style element.

Note
When *style-element* is used, only the style element’s COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTSIZE attributes are used.

Tip
You can use the **INSIDEVALUEATTRS=** option to change text attributes for values that are located inside the tiles.

**VALUELOCATION=**AUTO | INSIDE | OUTSIDE
specifies the location of the category column values in a two-way plot.

**AUTO**
locates the values for the second category in the CATEGORY= list that are outside of the tiles.
If the first category in the CATEGORY= list has any missing crossings with the second category, or if any of the second category values collide, then the values are located inside the tiles per the default fit policy.

See “CATEGORY=(column-list)” on page 642

INSIDE
for each category value, locates the values inside the largest tile.

OUTSIDE
locates the values outside of the tiles, in the plot area.

Default AUTO

Restriction This option applies to two-way plots only.

XVALUEFITPOLICY=ROTATE | NONE
specifies a policy for avoiding collisions along the width of the plot among category values that are outside of the tiles.

ROTATE
rotates the values if any of the values collide.

NONE
does not attempt to fit values that collide.

Default ROTATE

YVALUEFITPOLICY=NULL | ROTATEALWAYS
specifies a policy for avoiding collisions along the height of the plot among category values that are outside of the tiles.

NONE
does not attempt to fit values that collide.

ROTATEALWAYS
rotates the values regardless of whether any of the values collide.

Default NONE

Interaction This option is effective only when VALUELOCATION=OUTSIDE.

Details
A mosaic plot displays relative frequencies for categorical variables. Each crossing of the categorical values is represented by a tile. The area of each tile is proportional to the frequency of that crossing. The plot is the result of an iterative process. The first iteration splits the plot area into tiles along the width according to the relative frequency of the first category column values. Subsequent iterations split the tiles from the previous iteration in the direction orthogonal to the previous split by using the relative frequencies of each category column’s values. By default, the gap (or gutter) for each split gets progressively smaller, with a minimum gap of 3 pixels.

You can use the GUTTER= option to specify a different gap for each split.

The following figure provides an example of a three-way mosaic plot, which has three categories.
Note: The plot contains equally sized tiles for demonstration purposes.

In the example plot, the first split is along the width for CATEGORY 1. The second split is along the height for CATEGORY 2. Finally, the third split is along the width for CATEGORY 3. Notice how the gaps between the tiles get progressively smaller from the first split to the last split.

Example: MOSAICPLOTPARM Statement

The following graph was generated by the “Example Program” on page 653:
Example Program

/* Summarize the SASHELP.CARS data for ORIGIN and TYPE */
proc summary data=sashelp.cars nway;
  class origin type;
  var mpg_highway;
  output out=mileage mean=avgMpg N=count / noinherit;
run;

/* Generate the plot */
proc template;
  define statgraph mosaicPlotParm;
  begingraph;
  layout region;
  mosaicPlotParm category=(type origin) count=count /
    name="mosaic" colorresponse=avgMpg;
  continuouslegend "mosaic" / title="Average Miles Per Gallon"
    pad=(left=5);
  endlayout;
  endgraph;
end;
run;

proc sgrender data=mileage template=mosaicPlotParm;
run;
NEEDLEPLOT Statement
Creates a plot of observations as points connected to a baseline by vertical line segments.

Syntax

```
NEEDLEPLOT X=column | expression
    Y=numeric-column | expression </options>;
```

Summary of Optional Arguments

**Appearance options**
- `BASELINEATTRS=style-element | (line-options)`
  specifies the appearance of the baseline.
- `CLUSTERWIDTH=number`
  specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.
- `DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN`
  enhances the visual appearance of the needle plot lines.
- `DATATRANSPARENCY=number`
  specifies the degree of the transparency of the needle lines, markers, and data labels, if displayed.
- `DISPLAY=(display-options) | STANDARD | ALL`
  specifies which needle features to display.
- `DISPLAYBASELINE=ON | OFF | AUTO`
  specifies whether the baseline is displayed.
- `INDEX=positive-integer-column | expression`
  specifies indices for mapping line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.
- `LINEATTRS=style-element | style-element (line-options) | (line-options)`
  specifies the attributes of the needle lines for the data points.
- `MARKERATTRS=style-element | style-element (marker-options) | (marker-options)`
  specifies the attributes of the data markers.

**Axes options**
- `BASELINEINTERCEPT=number | RELATIVE`
  specifies the Y-intercept for the baseline.
- `PRIMARY=TRUE | FALSE`
  specifies that the data columns for this plot and the plot type be used for determining default axis features.
- `XAXIS=X | X2`
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
- `YAXIS=Y | Y2`
  specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.
Data tip options

**ROLENAME=(role-name-list)**
specifies user-defined roles that can be used to display information in the data tips.

**TIP=(role-list) | NONE**
specifies the information to display when the cursor is positioned over a needle line or marker.

**TIPFORMAT=(role-format-list)**
specifies display formats for tip columns.

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns.

Label options

**DATALABEL=column**
specifies labels at the data points.

**DATALABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the data labels.

**DATALABELPOSITION=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT**
specifies the location of the data labels relative to the end of the needle lines and markers, if displayed.

**DATALABELSPLIT=TRUE | FALSE**
specifies whether to split the data labels at the specified split characters.

**DATALABELSPLITCHAR="character-list"**
specifies one or more characters on which the data labels can be split if needed.

**DATALABELSPLITCHARDROP=TRUE | FALSE**
specifies whether the split characters are included in the data labels.

**DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**
specifies the justification of the strings that are inside the data label blocks.

**LEGENDLABEL="string"**
specifies a label to be used in a discrete legend for this plot.

Midpoint options

**DISCRETEOFFSET=number**
specifies an amount to offset all needle lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.

**GROUP=column | discrete-attr-var | expression**
creates a distinct set of needle lines, markers, and data labels for each unique group value of the specified column.

**GROUPDISPLAY=OVERLAY | CLUSTER**
specifies whether grouped needle lines are overlaid or clustered around the category midpoints on a discrete axis or around the intervals on an interval axis.

**GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING**
specifies the ordering of the groups within a category.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.
ODS options

URL=string-column
specifies an HTML page to display when a needle or marker is selected.

Plot reference options

NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments

X=column | expression
specifies a column or expression for the X values.

Y=numeric-column | expression
specifies a numeric column or numeric expression for the Y values.

Optional Arguments

BASELINEATTRS=style-element | (line-options)
specifies the appearance of the baseline.

Default
The COLOR= option in the LINEATTRS= option, if specified. Otherwise, the GraphDataDefault:contrastColor style reference.

Notes
The baseline is always drawn by default.

When style-element is specified, only the style element's COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip
To suppress the baseline, set the line thickness to 0:

baselineattrs=(thickness=0)

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

BASELINEINTERCEPT=number | RELATIVE
specifies the Y-intercept for the baseline. Prior to SAS 9.4M5, the baseline is always displayed in the chart, whether for a specified value or for the default value. Starting with SAS 9.4M5, the baseline display is controlled by the DISPLAYBASELINE= option, which is ON by default.

number
specifies the Y-intercept value to use for the baseline.

Interactions
When number is specified, the response axis data range is extended if necessary to include the baseline intercept. When a logarithmic response axis is requested and number is 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINEINTERCEPT= to a positive value.

When DISPLAYBASELINE=AUTO is in effect, the baseline is not displayed if the baseline intercept is equal to the minimum or maximum value of the response-axis range.
**Tips**

The baseline does not add a tick or a tick value to the axis. To label the baseline, you can overlay a REFERENCELINE statement with the same Y value and use its CURVELABEL option.

The appearance of the baseline is controlled by the BASELINEATTRS= option.

To suppress the baseline, prior to SAS 9.4M5, use the BASELINEATTRS= option to set the line thickness to 0. Starting with SAS 9.4M5, specify DISPLAYBASELINE=OFF.

---

**RELATIVE**

places the baseline at the Y-axis tick mark closest to the minimum of the range for the needle data points.

Default 0

**CLUSTERWIDTH=number**

specifies the width of the group clusters as a fraction of the midpoint spacing or bin width.

Default 0.85

Range 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Requirement For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

Interaction When markers are displayed for interval data and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

---

**DATALABEL=column**

specifies labels at the data points.

Default No data labels are displayed

Note The position of the labels is adjusted to prevent the labels from overlapping.

**DATALABELATTRS=style-element | style-element (text-options) | (text-options)**

specifies the color and font attributes of the data labels.

Defaults For non-grouped data, the GraphDataText style element.

For grouped data, the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

Interaction For this option to have any effect, the DATALABEL= option must also be specified.
When the DATALABELPOSITION=AUTO option is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

DATALABELPOSITION=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the end of the needle lines and markers, if displayed.

AUTO
uses a data-label collision avoidance algorithm to position the labels.

TOPRIGHT
TOP
TOPLEFT
LEFT
CENTER
RIGHT
BOTTOMLEFT
BOTTOM
BOTTOMRIGHT
specifies a location for all of the data labels.

Default AUTO

DATALABELSPLIT=TRUE | FALSE
specifies whether to split the data labels at the specified split characters. When set to TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters.

Default FALSE. The data labels are not split.

Requirement The DATALABEL= option must also be specified.

Interactions The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

This option has no effect when DATALABELPOSITION=AUTO.

See “boolean ” on page 1479 for other Boolean values that you can use.

DATALABELSPLITCHAR="character-list"
specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.
When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

“character-list”
one or more characters with no space between each character and enclosed in quotation marks.

Default
A blank space

Requirements
The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

datalabelsplitchar="abc"

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interactions
This option has no effect if DATALABELPOSITION=AUTO.

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

Notes
When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip
Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=TRUE | FALSE**
specifies whether the split characters are included in the data labels.

**TRUE**
drops the split characters from the data label.

**FALSE**
includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:

- the data label text for this label is Product*Group*A
- DATALABELSPLIT=TRUE
- DATALABELSPLITCHARDROP=TRUE | FALSE
- DATALABELSPLITCHAR=***
When DATALABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When DATALABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**

TRUE. The split characters are dropped from the data label.

**Requirement**

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction**

The DATALABELSPLITCHAR= option specifies the split characters.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**DATALABELSPLITJUSTIFY=**

AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the data label blocks.

**AUTO**

justifies the labels based on the DATALABELPOSITION= option.

<table>
<thead>
<tr>
<th>DATALABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

**CENTER**

LEFT

RIGHT

justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.

**Note:** The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.

In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For
example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

Default AUTO

Requirement The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interaction This option has no effect if DATALABELPOSITION=AUTO.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN** enhances the visual appearance of the needle plot lines. The following figure shows a needle plot with each of the skins applied.

![Needle Plot Skins](image)

Default The DATASKIN= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

Restriction Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

Interaction This option overrides the BEGINGRAPH statement DATASKIN= option.

**DATATRANSPARENCY=number** specifies the degree of the transparency of the needle lines, markers, and data labels, if displayed.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

**DISCRETEOFFSET=number** specifies an amount to offset all needle lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.
Default 0 (no offset, all needle lines and markers are centered on the discrete X values)

Range -0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is to the right. If the parent layout’s axis options set REVERSE=TRUE, then the offset direction is also reversed.

Restriction This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

Tip Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

DISPLAY=(display-options) | STANDARD | ALL specifies which needle features to display.

(display-options)
a space-separated list of one or more options enclosed in parentheses.

Currently, only the MARKERS option is supported, which displays needle lines with markers.

STANDARD displays needle lines without markers

ALL displays needle lines with markers

Default STANDARD

Tip Use the MARKERATTRS= and LINEATTRS= options to control the appearance of the line and markers.

DISPLAYBASELINE=ON | OFF | AUTO specifies whether the baseline is displayed.

ON always displays the baseline.

OFF does not display the baseline.

AUTO displays the baseline if the baseline intercept is within the response-axis range, excluding the minimum and maximum axis values.

This is the typical case when the needle plot contains both positive and negative needles, and the default baseline intercept of 0 is used. If the baseline intercept is equal to the minimum or maximum value of the response-axis range, the baseline is not displayed.

Default ON

Note: This feature applies to SAS 9.4M5 and to later releases.
Tip
By default, the baseline intercept is 0. Use the BASELINEINTERCEPT= option to change the baseline intercept.

GROUP=column | discrete-atr-var | expression
creates a distinct set of needle lines, markers, and data labels for each unique group value of the specified column.

discrete-atr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default
Each distinct group value might be represented in the plot by a different combination of color, line pattern, and marker symbol. These vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1–GraphDataN and GraphMissing style elements.

Interactions
The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of marker symbols, colors, and line patterns.

The marker size is set by the MARKERATTRS= option.

The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Notes
The representations that are used to identify the groups can be overridden. For example, each distinct group value might be represented by a different line pattern, but the LINEATTRS=(PATTERN=pattern) option could be used to assign the same line pattern to all of the plot’s line patterns, letting line color indicate group values. Likewise, LINEATTRS=(COLOR=color) could be used to assign the same color to all lines, letting line pattern indicate group values.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

See
"DISCRETEATTRVAR Statement" on page 1435

GROUPDISPLAY=OVERLAY | CLUSTER
specifies whether grouped needle lines are overlaid or clustered around the category midpoints on a discrete axis or around the intervals on an interval axis.

OVERLAY
centers the needle lines for matching category values on the midpoints.

The needle lines in each set of group values are superimposed on each other.
CLUSTER
clusters the needle lines for matching category values around the midpoints.
Each cluster of group values is centered at the midpoint for the category.

Default: OVERLAY

Interaction: For this option to take effect, the GROUP= option must also be specified.

Note: When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in the group-column data order.

REVERSEDATA
orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip: This option is useful when you want to reverse the category axis.

ASCENDING
orders the groups within a category in ascending order.

DESCENDING
orders the groups within a category in descending order.

Defaults: DATA if the data is a SAS data set.
ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

Interactions: This option is ignored if the GROUP= option is not also specified.
By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes: When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.
In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

| Tips | Use the CLUSTERWIDTH= option to control the distance between the group markers in a cluster. |
|      | Use the INDEX= option to alter the default sequence of visual attributes that is assigned to the groups. |

**INCLUDEDMISSINGGROUP**=TRUE | FALSE

specifies whether missing values of the group variable are included in the plot.

| Default | TRUE |
| Interaction | For this option to take effect, the GROUP= option must also be specified. |
| Tip | The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element. |
| See | “boolean ” on page 1479 for other Boolean values that you can use. |

**INDEX**=positive-integer-column | expression

specifies indices for mapping line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

| Requirements | The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored. |
| | The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored. |
| | All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable. |
| Interaction | For this option to take effect, the GROUP= option must also be specified. |
| Notes | The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use. |
| | If you do not use this option, then the group values are mapped in the order of the data. |
Tip
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL=’string’**
specifies a label to be used in a discrete legend for this plot.

Default The Y-column label. If a label is not defined, then the Y-column name is used.

Restriction This option applies only to an associated DISCRETELEGEND statement.

Interaction If the GROUP= option is specified, then this option is ignored.

**LINEATTRS=style-element | style-element (line-options) | (line-options)**
specifies the attributes of the needle lines for the data points.

Defaults For non-grouped data, the GraphDataDefault style element.

For grouped data, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the LineThickness attribute of the GraphDataDefault style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**MARKERATTRS=style-element | style-element (marker-options) | (marker-options)**
specifies the attributes of the data markers.

Defaults For non-grouped data, GraphDataDefault style element.

For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interactions This option’s COLOR= suboption overrides the default behavior for grouped data. When the COLOR= suboption is specified in that case, all markers have the same color, and the marker symbol alone distinguishes the markers.

This option’s SYMBOL= suboption overrides the default behavior for grouped data. When the SYMBOL= suboption is specified in that case, all markers have the same symbol, and the symbol color alone distinguishes the markers.

The TRANSPARENCY= fill option overrides this option’s DATATRANSPARENCY= suboption.

This option is ignored if the DISPLAY= option disables the display of the markers.

If the DATASKIN= option is in effect, then the data skin determines the marker outlines. Any outline-related settings from the current ODS style or from the marker attribute options are ignored.
Note When style-element is specified, only the style element’s MARKERSYMBOL, CONTRASTCOLOR, and MARKERSIZE attributes are used.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Marker Options” on page 1523 for available marker-options.

**NAME=“string”**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**PRIMARY=TRUE | FALSE**

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

(role-name-list) a space-separated list of role-name = column pairs.

Example The following example assigns the column Obs to the user-defined role TIP:

```
ROLENAME=(TIP1=OBS)
```

Default No user-defined roles

Requirement The role names that you choose must be unique and different from the predefined roles X, Y, DATALABEL, INDEX, and GROUP.
TIP=\(\text{role-list}\) | NONE

specifies the information to display when the cursor is positioned over a needle line or marker. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the needle plot can be specified along with roles that do.

\(\text{role-list}\)

an ordered, space-separated list of unique NEEDLEPLOT and user-defined roles.

NEEDLEPLOT roles include \(X\), \(Y\), DATALABEL, and GROUP.

User-defined roles are defined with the \texttt{ROLENAME=} option.

Example

The following example displays data tips for the columns assigned to the roles \(X\) and \(Y\) as well as the column Obs, which is not assigned to any pre-defined NEEDLEPLOT role. The Obs column must first be assigned a role.

\begin{verbatim}
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
\end{verbatim}

NONE

suppresses data tips and URLs (if requested) from the plot.

Default

The columns assigned to these roles are automatically included in the data tip information: \(X\), \(Y\), DATALABEL, and GROUP.

Requirement

To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip

The labels and formats for the TIP roles can be controlled with the \texttt{TIPLABEL=} and \texttt{TIPFORMAT=} options.

TIPFORMAT=\(\text{role-format-list}\)

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

\begin{verbatim}
(\text{role-format-list})
\end{verbatim}

a space-separated list of \texttt{role-name} = \texttt{format} pairs.

Example

\begin{verbatim}
ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
\end{verbatim}

Default

The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction

Only the roles that appear in the TIP= option are used.

Requirement

A column must be assigned to each of the specified roles. (See the \texttt{ROLENAME=} option.)
**TIPLABEL=(role-label-list)**

specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

*role-label-list*

a space-separated list of rolename ="string" pairs.

**Example**

```plaintext
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

**Default**

The column label or column name of the column assigned to the role.

**Restriction**

Only the roles that appear in the TIP= option are used.

**Requirement**

A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**URL=string-column**

specifies an HTML page to display when a needle or marker is selected.

*string-column*

specifies a column that contains a valid HTML page reference (HREF) for each needle that is to have an active link.

**Example**

```plaintext
http://www.sas.com
```

**Requirement**

To generate selectable needle lines, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interactions**

This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tips**

The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding needle or marker is selected.

The URL value can be the same for any X and Y pairs. In that case, the same action is taken when the needle or marker is selected for those X and Y pairs.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default**

X
Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS= Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.
Default: Y

Details
In the NEEDLEPLOT statement, the X column can specify character or numeric values. The Y column must specify numeric values. For character columns, the X axis is always of TYPE=DISCRETE. For numeric columns, the X axis is of TYPE=LINEAR by default.
The Y axis is of TYPE=LINEAR by default.

Example: NEEDLEPLOT Statement
The following graph was generated by the “Example Program” on page 670:

Example Program
```
proc template;
```

---

IBM Stock Trend

- Y-axis: Price per Share
  - Range: $60 to $120

- X-axis: Date
  - From January 2000 to January 2006

---

Example Program
```
proc template;
define statgraph needleplot;
begingraph;
    entrytitle "IBM Stock Trend";
    layout overlay;
    needleplot x=date y=close / baselineintercept=80 lineattrs=(color=blue);
    endlayout;
    endgraph;
end;
run;

proc sgrender data=sashelp.stocks template=needleplot;
    where stock="IBM" and date > "31dec1999"d;
run;

PBSLINEPLOT Statement

Creates a fitted penalized B-spline curve computed from input data.

Restriction: The PBSLINEPLOT statement supports only models of one independent and one dependent variable.

Tip: Starting with SAS 9.4M3, subpixel rendering is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

PBSLINEPLOT X=numeric-column | expression
Y=numeric-column | expression /[<regression-options> <options>];

Summary of Optional Arguments

Appearance options

DATATRANSPARENCY=number
    specifies the degree of the transparency of the curve and curve label.

INDEX=positive-integer-column | expression
    specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

LINEATTRS=style-element | style-element (line-options) | (line-options)
    specifies the line attributes of the spline curve.

Axes options

PRIMARY=TRUE | FALSE
    specifies that the data columns for this plot and the plot type be used for determining default axis features.

XAXIS=X | X2
    specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
YAXIS=\texttt{Y | Y2}
   specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

\texttt{TIPFORMAT=(role-format-list)}
   specifies display formats for tip columns.

\texttt{TIPLABEL=(role-label-list)}
   specifies display labels for tip columns.

Label options

\texttt{CURVELABEL="string"}
   specifies a label for the spline curve.

\texttt{CURVELABELATTRS=style-element \, style-element (text-options) \, (text-options)}
   specifies the color and font attributes of the spline curve labels.

\texttt{CURVELABELLOCATION=INSIDE \, OUTSIDE}
   specifies the location of the spline curve label relative to the plot area.

\texttt{CURVELABELPOSITION=AUTO \, MAX \, MIN \, START \, END}
   specifies the position of the spline curve label relative to the curve line.

\texttt{CURVELABELSPLIT=TRUE \, FALSE}
   specifies whether to split the curve label at the specified split characters.

\texttt{CURVELABELSPLITCHAR="character-list"}
   specifies one or more characters on which the curve label can be split if needed.

\texttt{CURVELABELSPLITCHARDROP=TRUE \, FALSE}
   specifies whether the split characters are included in the curve label text.

\texttt{CURVELABELSPLITJUSTIFY=AUTO \, CENTER \, LEFT \, RIGHT}
   specifies the justification of the strings that are inside the curve label block.

\texttt{LEGENDLABEL="string"}
   specifies a label to be used in a discrete legend for this plot.

Midpoint options

\texttt{GROUP=column \, discrete-attr-var \, expression}
   creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

\texttt{INCLUDEMISSINGGROUP=TRUE \, FALSE}
   specifies whether missing values of the group variable are included in the plot.

Plot reference options

\texttt{NAME="string"}
   assigns a name to this plot statement for reference in other template statements.

Required Arguments

\texttt{X=numeric-column \, expression}
   specifies the column for the X values.

\texttt{Y=numeric-column \, expression}
   specifies the column for the Y values.
Optional Arguments

**CURVELABEL="string"**
specifies a label for the spline curve.

<table>
<thead>
<tr>
<th>Default</th>
<th>No curve label is displayed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>This option is not valid when the GROUP= option is specified.</td>
</tr>
<tr>
<td>Tip</td>
<td>The font and color attributes for the label are specified by the CURVELABELATR�= option.</td>
</tr>
</tbody>
</table>

**CURVELABELATR�s=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the spline curve labels.

<table>
<thead>
<tr>
<th>Default</th>
<th>The GraphValueText style element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactions</td>
<td>For this option to take effect, the CURVELABEL= option must also be specified.</td>
</tr>
<tr>
<td></td>
<td>If the GROUP= option is specified, then this option is ignored.</td>
</tr>
<tr>
<td>See</td>
<td>&quot;General Syntax for Attribute Options&quot; on page 1519 for the syntax on using a style-element.</td>
</tr>
<tr>
<td></td>
<td>&quot;Text Options&quot; on page 1525 for available text-options.</td>
</tr>
</tbody>
</table>

**CURVELABELLOCATION=INSIDE | OUTSIDE**
specifies the location of the spline curve label relative to the plot area.

**INSIDE**
locates the labels inside the plot area

**OUTSIDE**
locates the labels outside the plot area

<table>
<thead>
<tr>
<th>Default</th>
<th>INSIDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>OUTSIDE cannot be used when the PBSPLINEPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.</td>
</tr>
<tr>
<td>Interactions</td>
<td>For this option to take effect, the CURVELABEL= option must also be specified.</td>
</tr>
<tr>
<td></td>
<td>This option is used in conjunction with the CURVELABELPOSITION= option to determine where the curve labels appear.</td>
</tr>
<tr>
<td>See</td>
<td>&quot;Location and Position of Curve Labels&quot; on page 200</td>
</tr>
</tbody>
</table>

**CURVELABELPOSITION=AUTO | MAX | MIN | START | END**
specifies the position of the spline curve label relative to the curve line.

**AUTO**
automatically positions the spline curve label near the curve boundary along unused axes whenever possible (typically Y2 and X2) in order to avoid collision with tick values.
Restriction  This option is used only when CURVELABELPOSITION=OUTSIDE.

<table>
<thead>
<tr>
<th>MAX</th>
<th>forces the spline curve label to appear near maximum curve values (typically, upper right)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN</td>
<td>forces the spline curve label to appear near minimum curve values (typically, lower left)</td>
</tr>
<tr>
<td>START</td>
<td>forces the spline curve label to appear near the beginning of the curve.</td>
</tr>
<tr>
<td><strong>Restriction</strong></td>
<td>This option is used only when CURVELABELLOCATION=INSIDE.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>This option is particularly useful when the curve line has a spiral shape.</td>
</tr>
</tbody>
</table>

| END            | forces the spline curve label to appear near the end of the curve.                       |
| **Restriction**| This option is used only when CURVELABELLOCATION=INSIDE.                                  |
| **Tip**        | This option is particularly useful when the curve line has a spiral shape.                |

**Defaults**

<table>
<thead>
<tr>
<th>AUTO when CURVELABELLOCATION=OUTSIDE.</th>
</tr>
</thead>
<tbody>
<tr>
<td>END when CURVELABELLOCATION=INSIDE.</td>
</tr>
</tbody>
</table>

**Interactions**

For this option to take effect, the CURVELABEL= option must also be specified. The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

This option is used in conjunction with the CURVELABELLOCATION= option to determine where the spline curve label appears.

**Note**

When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the spline curve label might fall outside of the graph area. In that case, the spline curve label might not be displayed or might be positioned incorrectly.

**Tip**

When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.

**See**

"Location and Position of Curve Labels" on page 200
CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the curve label at the specified split characters. When a
curve label is split, the label is split on each occurrence of the specified split
characters.

Default FALSE. The curve label is not split.

Requirement The CURVELABEL= option must also be specified.

Interactions The CURVELABELSPLITCHAR= option specifies one or more
characters on which the splits occur.

This option has no effect when CURVELABELPOSITION=AUTO.

See “boolean " on page 1479 for other Boolean values that you can
use.

CURVELABELSPLITCHAR="character-list"
specifies one or more characters on which the curve label can be split if needed.
When multiple split characters are specified, each character in the list is treated
as a separate split character unless the specified characters appear
consecutively in the curve label. In that case, all of the specified split characters
together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the curve
label is split unconditionally at each occurrence of any of the specified split
characters. If the curve label does not contain any of the specified characters,
then the label is not split.

"character-list"
one or more characters with no delimiter between each character and
enclosed in quotation marks.

Default A blank space

Requirements The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no delimiters. For
example, to specify the split characters a, b, and c, use the
following option:

curvelabelsplitchar="abc"

The CURVELABEL= option and the CURVELABELSPLIT=TRUE
option must also be specified.

Interactions This option has no effect if CURVELABELPOSITION=AUTO.

The CURVELABELSPLITCHARDROP= option specifies whether
the split characters are included in the curve label or are dropped.

Notes When multiple characters are specified, the order of the
characters in the list is not significant.

The split characters are case sensitive.

Tip Use the CURVELABELSPLITJUSTIFY= option to specify the
justification of the strings in the curve label block.
CURVELABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the curve label text.

TRUE
drops the split characters from the curve label text.

FALSE
includes the split characters in the curve label text.

When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a curve label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

Note: The horizontal line to the left of the label represents the maximum end of the curve for reference.

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default TRUE. The split characters are dropped from the curve label.

Requirement The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction The CURVELABELSPLITCHAR= option specifies the split characters.

See “boolean ” on page 1479 for other Boolean values that you can use.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the curve label block.

AUTO
justifies the labels based on the CURVELABELPOSITION= option.
**CURVELABELPOSITION= Value Justification**

<table>
<thead>
<tr>
<th>Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

**CENTER**

**LEFT**

**RIGHT**

generates the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

---

**Note:** The horizontal line to the left of each label represents the maximum end of the curve for reference.

---

<table>
<thead>
<tr>
<th>AUTO Product Group A</th>
<th>CENTER Product Group A</th>
<th>LEFT Product Group A</th>
<th>RIGHT Product Group A</th>
</tr>
</thead>
</table>

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

**Default** AUTO

**Requirement** The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction** This option has no effect if CURVELABELPOSITION=AUTO.

**DATATRANSPARENCY=number**

specifies the degree of the transparency of the curve and curve label.

**Default** 0

**Range** 0–1, where 0 is opaque and 1 is entirely transparent

**GROUP=column | discrete-attr-var | expression**

creates a distinct set of curves from just the observations that correspond to each unique group value of the specified column.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction** A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.
Each distinct group value might be represented in the plot by a different combination of color and line pattern. Line colors vary according to the ContrastColor attribute of the GraphData1–GraphDataN and GraphMissing style elements. Line patterns vary according to the LineStyle attribute of the GraphData1–GraphDataN and GraphMissing style elements.

Starting with SAS 9.4M5, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID.

You must sort the data in ascending order by the specified group column before you plot the data. If the plot data is a CAS in-memory table, you must download the in-memory table to your SAS client, and then sort it locally before you plot it.

The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line colors and line patterns.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

The LINEATTRS= option can be used to override the representations that are used to identify the groups. For example, LINEATTRS=(PATTERN=SOLID) can be used to assign the same pattern to all of the loess curves, letting the line color distinguish group values. Likewise, LINEATTRS=(COLOR=BLACK) can be used to assign the same color to all of the curves, letting the line pattern distinguish group values.

The INCLUDEMISSINGGROUP=TRUE | FALSE specifies whether missing values of the group variable are included in the plot.

Default TRUE

For this option to take effect, the GROUP= option must also be specified.

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the
missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See “boolean ” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**

specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements**
The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Notes**
The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip**
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL="string"**

specifies a label to be used in a discrete legend for this plot.

**Default**
The string specified on the NAME= option.

**Restriction**
This option applies only to an associated DISCRETELEGEND statement.

**Interaction**
If the GROUP= option is specified, then this option is ignored.

**LINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the line attributes of the spline curve.

**Default**
The GraphFit style element.

**Restriction**
Starting with **SAS 9.4M5**, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID.
NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction The string is used as the default legend label if the LEGENDLABEL= option is not used.

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restrictions This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979
“boolean ” on page 1479 for other Boolean values that you can use.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example TIPFORMAT=(Y=6.2)

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Requirement To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Note The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the data tip information.
TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the
labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example   TIPLABEL=(Y="Curve")

Default   The column label or column name of the column assigned to the role.

Note      The columns assigned to the X, Y, and GROUP (if assigned) roles are
automatically included in the data tip information.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the
secondary X2 (top) axis.

Default   X

Interaction The overall plot specification and the layout type determine the axis
display. For more information, see “How Axis Features Are
Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the
secondary Y2 (right) axis.

Default   Y

Interaction The overall plot specification and the layout type determine the axis
display. For more information, see “How Axis Features Are
Determined” on page 973.

PBSPLINE Regression Options

ALPHA=positive-number
specifies the confidence level to compute.

Default   0.05

Range     0 < positive-number < 1

Tip       ALPHA=0.05 represents a 95% confidence level.

CLI="name"
produces confidence limits for individual predicted values for each observation.

Interaction name is a unique name within the template that is case sensitive and
cannot contain spaces. It must be assigned in order for the
confidence limits to be computed. To display confidence limits, you
must use this name as the required argument of a MODELBAND
statement. See the example in the section “Example:
PBSPLINEPLOT Statement” on page 683.

CLM="name"
produces confidence limits for a mean predicted value for each observation.
Interaction name is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBAND statement. See the example in the section "Example: PBSPLINEPLOT Statement" on page 683.

**DEGREE=non-negative-integer**
specifies the degree of B-spline.

- Default: 3
- Ranges: 0–174 in SAS 9.4M1 and earlier releases.
- Ranges: 0–10 starting with SAS 9.4M2.
- Restriction: Starting with SAS 9.4M2, DEGREE= and NKNOTS= cannot be set to 0 simultaneously. When both are set to 0, an error results.

**FREQ=numeric-column**
specifies a column in the input data set that represents the frequency of occurrence of the current observation, essentially treating the data set as if each observation appeared $n$ times, where $n$ is the value of the FREQ column for the observation. Noninteger values of the FREQ column are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ column is greater than or equal to 1.

**MAXPOINTS=positive-integer**
specifies the maximum number of predicted points generated for the spline curve as well as any confidence limits.

- Default: 201

**NKNOTS=non-negative-integer**
specifies the number of evenly spaced internal knots. By default, a large number of knots (100) is specified, which allows for an extreme lack of smoothness in the results. However, the final function is typically much smoother due to the penalty. When SMOOTH=0 is specified, you should typically ask for many fewer knots than the default, since there is no penalty for lack of smoothness. For example, ten or fewer knots is usually enough to follow the functional form found in most data.

- Default: 100
- Restriction: Starting with SAS 9.4M2, MKNOTS= and DEGREE= cannot be set to 0 simultaneously. When both are set to 0, an error results.

See example “Penalized B-Splines” in the TRANSREG procedure description in SAS/STAT User's Guide.

**SMOOTH=AUTO | non-negative-number**
specifies a regression parameter value.

- Default: AUTO
- Note: With SMOOTH=AUTO, a regression parameter that minimizes a lack-of-smoothness penalty is automatically selected.
Tip You can specify SMOOTH=0 to get an ordinary B-spline fit.

**WEIGHT=numeric-column**
specifies a column in the input data set that contains values to be used as a priori weights for a penalized B-spline fit. If an observation’s weight is zero, negative, or missing, then the observation is deleted from the analysis.

Interaction Starting with SAS 9.4M2, when the CLI= option is used with this option, the confidence band for individual predicted values is displayed as a high-low chart instead of a band.

Details

The PBSPLINEPLOT statement supports only models of one independent and one dependent variable. For more information about the fitting methodology, see the TRANSREG procedure in the SAS/STAT user’s guide.

In addition to the penalized B-spline, the PBSPLINEPLOT statement can compute confidence levels for the fitted line. To display the confidence levels,

1. use the CLI= or CLM= option to declare a name for the confidence level
2. use a MODELBAND statement to refer to this name. This statement draws a confidence band from this information. See “MODELBAND Statement” on page 629 for information about how to control the appearance of the confidence band.

Example: PBSPLINEPLOT Statement

The following graph was generated by the “Example Program” on page 684:
Example Program

```plaintext
proc template;
  define statgraph pbsplineplot;
  begingraph;
    entrytitle "Spline Fit";
    layout overlay;
      scatterplot x=weight y=mpg_highway / datatransparency=0.7;
      pbsplineplot x=weight y=mpg_highway / name="fitline"
        alpha=0.05 legendlabel="Spline Fit";
    discretelegend "fitline";
  endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template=pbsplineplot;
run;
```

PIECHART Statement

Creates a pie chart that is computed from input data.

Requirement: The PIECHART statement must be placed in a LAYOUT REGION, LAYOUT GRIDDED, or LAYOUT LATTICE block. It cannot be placed in an overlay-type layout such as LAYOUT OVERLAY or LAYOUT OVERLAYEQUATED because a pie chart does not have axes.

Note: The PIECHART statement does not honor the ODS GRAPHICS options DISCRETEMAX=, GROUPMAX=, and LABELMAX=.

Syntax

```
PIECHART CATEGORY=column | discrete-attr-var | expression <options>;
PIECHART CATEGORY=column | discrete-attr-var | expression
  RESPONSE=numeric-column | expression <options>;
```

Summary of Optional Arguments

Appearance options

```
CATEGORYDIRECTION=COUNTERCLOCKWISE | CLOCKWISE
  specifies whether to display the pie slices in counterclockwise or clockwise sequence.
CENTERFIRSTSLICE=TRUE | FALSE
  specifies whether the first pie slice is centered on the starting angle or starts on the starting angle.
DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of the filled pie slices.
DATATRANSPARENCY=number
```
specifies the degree of the transparency of all pie slices, outlines, and text.

**DISPLAY=(display-options) | STANDARD**
specifies which pie-slice options to display.

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the filled pie slices.

**OTHERSLICEOPTS=(other-slice-options)**
specifies the properties of the Other slice.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**
specifies the properties of the pie and slice outlines.

**START=degrees**
specifies which degree between 0 and 360 serves as the starting position for the first pie slice.

**Data tip options**

**TIP=(role-list) | NONE**
specifies the information to display when the cursor is positioned over a pie slice.

**TIPFORMAT=(role-format-list)**
specifies display formats for tip columns.

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns.

**Grouping options**

**GROUP=column | expression**
creates a separate concentric annulus (or stacked cylinders) for each unique group value of the specified column.

**GROUPGAP=dimension**
specifies a dimension for the optional gap that can be displayed between each annulus of a grouped pie.

**GROUPLABELOPTS=(grouplabel-options)**
specifies text attributes, location, and other options for displaying group labels.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the Group column are included in the pie.

**OTHERSLICE=TRUE | FALSE**
specifies whether to consolidate smaller pie slices into a single slice that represents “other” values that are in the data, or whether to display those smaller slices as separate pie slices.

**Label options**

**DATALABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the slice labels.

**DATALABELCONTENT=ALL | STANDARD | NONE | (content-options)**
specifies the information to display in the slice labels.

**DATALABELLOCATION=AUTO | INSIDE | OUTSIDE | CALLOUT**
specifies whether to display the slice labels within the pie slices or outside of the pie circumference.

**LABELFITPOLICY=NONE | DROP**
specifies the label fitting policy to be used if a particular label does not fit within the pie slice.
ODS options
   URL=string-column
   specifies an HTML page to display when a pie slice is selected.

Plot reference options
   NAME="string"
   assigns a name to this plot statement for reference in other template
   statements.

Statistics options
   STAT=FREQ | PCT | SUM | MEAN
   specifies the statistic to be computed.

Required Argument
   CATEGORY=column | discrete-attr-var | expression
   specifies the column for the category values. Duplicated values of CATEGORY
   are summarized into a unique value. All values are treated as discrete.
   
   discrete-attr-var
   specifies a discrete attribute map variable.
   
   The discrete attribute map variable is created by a DISCRETEATTRVAR
   statement.
   
   Restriction A discrete attribute map variable specification must be a direct
   reference to the attribute map variable. It cannot be set by a dynamic variable.

   See "DISCRETEATTRVAR Statement" on page 1435

Optional Arguments
   CATEGORYDIRECTION=COUNTERCLOCKWISE | CLOCKWISE
   specifies whether to display the pie slices in counterclockwise or clockwise
   sequence.
   
   Default COUNTERCLOCKWISE
   
   Tip The START= option controls the starting angle for the first pie slice.

   CENTERFIRSTSLICE=TRUE | FALSE
   specifies whether the first pie slice is centered on the starting angle or starts on
   the starting angle.

   Note: This option is valid starting in SAS 9.4M1.

   The following figure shows the effect of this option on a pie chart in which Asia is
   the first category slice, the starting angle is 0 degrees, and the category direction
   is counterclockwise.
Default: FALSE

Tips

- Use the START= option to change the starting angle.
- Use the CATEGORYDIRECTION= option to change the category direction.

See

- "boolean " on page 1479 for other Boolean values that you can use.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)

- specifies the color and font attributes of the slice labels.

Default: The GraphValueText style element.

See

- "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.
- "Text Options" on page 1525 for available text-options.

DATALABELCONTENT=ALL | STANDARD | NONE | (content-options)

- specifies the information to display in the slice labels.

ALL
- displays all available information

STANDARD
- equivalent to specifying the two content-options CATEGORY and RESPONSE

NONE
- does not display slice labels

(content-options)
- a space-separated list of one or more of the following options enclosed in parentheses:
  
  CATEGORY
- displays the CATEGORY value

  PERCENT
- displays the PERCENT value when STAT=FREQ or STAT=PCT. When STAT=MEAN or STAT=SUM, it displays nothing

  RESPONSE
- displays the statistic that is requested in the STAT= option.
Defaults  
When `STAT=PCT`, the default is (CATEGORY PERCENT).

Otherwise, the default is STANDARD.

Note  
The position of the labels is adjusted to prevent the labels from overlapping.

\[
\text{DATALABELLOCATION=\text{AUTO} | \text{INSIDE} | \text{OUTSIDE} | \text{CALLOUT}}
\]

specifies whether to display the slice labels within the pie slices or outside of the pie circumference.

<table>
<thead>
<tr>
<th>DATALABELLOCATION=INSIDE</th>
<th>DATALABELLOCATION=CALLOUT</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Pie Chart INSIDE" /></td>
<td><img src="image" alt="Pie Chart CALLOUT" /></td>
</tr>
</tbody>
</table>

**AUTO**
automatically selects either INSIDE, OUTSIDE, or CALLOUT to optimize the label position

**INSIDE**
locates the slice labels inside the pie slices.

\[\text{Note: If a particular label does not fit within the pie slice, then the fit policy takes effect (set by the LABELFITPOLICY= option).}\]

**OUTSIDE**
locates the slice labels outside of the pie circumference.

**CALLOUT**
locates the slice labels outside of the pie circumference and draws a line from the label to its slice.

**Default**  AUTO

**Note**  When OUTSIDE or CALLOUT is specified and there is not enough space to display one or more of the labels, the labels are suppressed.

\[
\text{DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN}
\]

enhances the visual appearance of the filled pie slices. The following figure shows pies with each of the skins applied.
The DATASKIN= option value that is specified in the BEGINGRAPH statement. If not specified, then the GraphSkins:DataSkin style element value is used.

Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

For this option to have any effect, DISPLAY= FILL must be in effect. Otherwise, this option is ignored.

This option overrides the BEGINGRAPH statement DATASKIN= option.

The appearance of the data skin is based on the FILLATTRS= color.

When a data skin is applied, all slice outlines are set by the skin, and the OUTLINEATTRS= option is ignored.

**DATATRANSPARENCY=** *number*

specifies the degree of the transparency of all pie slices, outlines, and text.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

The FILLATTRS= option can be used to set transparency for just the pie slices. The OTHERSLICEOPTS= option can be used to specify transparency for the "other" slice. You can combine this option with FILLATTRS= and with OTHERSLICEOPTS= to set one transparency for the outlines and text but a different transparency for the pie slices.

Example:

datatransparency=0.2 fillattrs=(transparency=0.6)
**DISPLAY=** *(display-options) | STANDARD*

specifies which pie-slice options to display.

*(display-options)*

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**OUTLINE**

displays the pie-slice outline.

The default outline properties are set by the GraphOutline style element.

**FILL**

displays the pie-slice fill color.

The default colors are set by the Color attribute of the GraphData1–GraphDataN style elements. The fill color of the “other” slice (if shown) is from the color attribute of the GraphOther style element. If FILL is not specified, then an opaque pie is drawn using the background color of the containing layout.

**STANDARD**

specifies OUTLINE and FILL

**Default**

STANDARD

**Tip**

Use the **OUTLINEATTRS=** and **FILLATTRS=** options to control the appearance of the pie slices.

**FILLATTRS=** *(style-element | style-element (fill-options) | (fill-options))*

specifies the appearance of the filled pie slices. Prior to SAS 9.4M3, this option specifies the color and transparency of all of the pie slices, excluding the Other slice. Starting with SAS 9.4M3, this option specifies the transparency for the Other slice as well.

**Default**

The GraphDataDefault:Color style reference.

**Interaction**

For this option to have any effect, the fill must be enabled by the ODS style or the **DISPLAY=** option.

**Tips**

The FILLATTRS= suboption of the OTHERSLICEOPTS= option specifies the color and transparency of the Other slice.

The **DATATRANSPARENCY=** option sets the transparency for all pie slices, outlines, and text. You can combine this option with DATATRANSPARENCY= to set one transparency for the outlines and text but a different transparency for the pie slices. Example:

```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax for using a **style-element** value.

“Fill Color Options” on page 1520 for available **fill-options** values.

**GROUP=** *(column | expression)*

creates a separate concentric annulus (or stacked cylinders) for each unique group value of the specified column. The grouped rings are displayed in data order.
When this option is used, the unique column values are found and then the slice colors are taken from the GraphData1–GraphDataN style elements. Missing values in the data can affect the group order. You can use the INCLUDEMISSINGGROUP= option to manage missing group values. In addition, you can use INCLUDEMISSINGDISCRETE=TRUE in the BEGINGRAPH statement to create pie slices for missing CATEGORY values.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

This option creates only rings of pies. To create a grid of pies, specify the PIECHART statement within a LAYOUT LATTICE, LAYOUT DATALATTICE, or LAYOUT DATAPANEL statements.

GROUPGAP=dimension

specifies a dimension for the optional gap that can be displayed between each annulus of a grouped pie.

Default 0

For this option to take effect, the GROUP= option must also be specified.
Interaction

If the specified *dimension* is too large for the area that is available to the pie chart, then the results might be unexpected.

**Note**

The size of the inner pie remains the same regardless of the GROUPGAP= value.

**See**

“dimension” on page 1480

**GROUPLABELOPTS=**(*grouplabel-options*)

specifies text attributes, location, and other options for displaying group labels. The following *grouplabel-options* are available. One or more options can be specified as space-separated *name = value* pairs.

**LABEL=**AUTO | NONE | “*string*”

specifies a descriptive label for the Group column

**AUTO**

specifies the column label of the GROUP= column or the column name of the GROUP= column, if no column label exists.

**NONE**

specifies that no label is displayed

“*string*”

specifies a string to use as the label

**Default** AUTO

**LABELATTRS=***style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the text properties of the group label.

**Default** The GraphLabelText style element.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Text Options” on page 1525 for available *text-options*.

**LOCATION=**RIGHT | LEFT

specifies whether the block of text for group labeling appears to the right or left of the pie.

**Default** RIGHT

**VALUEATTRS=***style-element* | *style-element* (*text-options*) | (*text-options*)

specifies the text properties of the group values.

**Default** The GraphValueText style element.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Text Options” on page 1525 for available *text-options*.

**Default** The group label and values are shown as a block of text to the right or left of the pie. Slice labels are moved to the inside of the pie slices. A line is drawn from each group value to its annulus (or cylinder).
Restriction  For this option to take effect, the `GROUP=` option must also be specified.

**INCLUDEMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the Group column are included in the pie.

Default  TRUE

Restriction  For this option to take effect, the `GROUP=` option must also be specified.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**LABELFITPOLICY=NONE | DROP**

specifies the label fitting policy to be used if a particular label does not fit within the pie slice.

**NONE**

draws each label regardless of whether it fits within the slice region.

**DROP**

drops labels that do not fit within the slice region, but draws labels that do fit.

Default  NONE

Tip  This option determines how labels are managed when `DATALABELLOCATION= INSIDE` and a particular label does not fit within the pie slice.

**NAME= ”string”**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions  The `string` is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**OTHERSLICE=TRUE | FALSE**

specifies whether to consolidate smaller pie slices into a single slice that represents “other” values that are in the data, or whether to display those smaller slices as separate pie slices. If this option is set to FALSE, then all unique category values appear as slices. If this option is set to TRUE, then some of the smaller slices might be combined into a single slice, referred to as the Other slice.

Default  TRUE

Tip  To set the properties of the “other” slice, use the OTHERSLICEOPTS= option.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**OTHERSLICEOPTS=(other-slice-options)**

specifies the properties of the Other slice. Example:

```plaintext
piechart category=region / name="p"
```
The following *other-slice-options* values are available. You can specify one or more options as space-separated \texttt{name = value} pairs.

\textbf{TYPE=}\texttt{PERCENT | MAXSLICES}

specifies which method to use to determine the size of the Other slice.

\texttt{PERCENT}

uses the percentage that is set by the \texttt{PERCENT=} suboption.

\texttt{MAXSLICES}

uses the count that is set by the \texttt{MAXSLICES=} suboption.

**Default** \texttt{PERCENT}

\textbf{MAXSLICES=}\texttt{positive-integer}

specifies the maximum number of category values to represent with pie slices.

Any remaining values are consolidated into the Other slice.

**Default** \texttt{10}

**Interactions** For this option to have any effect, \texttt{TYPE=}\texttt{MAXSLICES} must also be specified among the suboptions for \texttt{OTHERSLICEOPTS=}. The slices are counted in the order in which they are displayed. This order is affected by the \texttt{CATEGORYDIRECTION=} option.

\texttt{PERCENT=}\texttt{percent-of-total}

collects all category values with response values less than or equal to the specified \texttt{percent-of-total} value into the Other slice.
4. Any original slice that represents 4% or less of the total is put in the Other category.

**Range**
0–100

**Interaction**
For this option to have any effect, TYPE=PERCENT must also be specified among the suboptions for OTHERSLICEOPTS=.

**LABEL=”string”**
specifies a label for the Other slice.

**Default**
"OTHER"

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the area fill for the Other slice.

This option does not affect the appearance of the area fill for the remaining slices.

**Default**
The GraphOther style element.

**Interactions**
Suboption TRANSPARENCY= in this option overrides the DATATRANSPARENCY= option for the other slice only.

Starting with SAS 9.4M3, suboption TRANSPARENCY= in this option overrides suboption TRANSPARENCY= in option FILLATTRS= in the PIECHART statement only for the Other slice.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**
specifies the properties of the pie and slice outlines.

**Default**
The GraphOutlines style element.

**Interactions**
For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

If the DATASKIN= option applies a data skin, then this option is ignored.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element value.

“Fill Color Options” on page 1520 for available fill-options values.

**START=degrees**
specifies which degree between 0 and 360 serves as the starting position for the first pie slice. A value of 0 degrees corresponds to the three o’clock position.

Degrees can be either positive or negative. Positive values move the starting position counterclockwise, and negative values move the starting position
clockwise. From the starting point specified by this option, the slices are drawn in the direction specified by the `CATEGORYDIRECTION=` option.

Default 0

Range 0 to 360

**STAT=**FREQ | PCT | SUM | MEAN

specifies the statistic to be computed. For pie charts with no `RESPONSE=` column:

FREQ frequency count

PCT percentages between 0 and 100

For pie charts with a `RESPONSE=` column:

SUM

MEAN

Defaults

SUM for pie charts that specify the `RESPONSE=` argument

FREQ for pie charts that do not specify the `RESPONSE=` argument

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over a pie slice. If this option is used, then it replaces all of the information that is displayed by default.

*(role-list) an ordered, space-separated list of unique PIECHART roles.*

PIECHART roles include CATEGORY, RESPONSE, and GROUP. The RESPONSE role represents the computed statistic for the CATEGORY value, based on the statistic that is set by the `STAT=` option.

**Example** The following example displays data tips for the columns assigned to the roles CATEGORY and RESPONSE.

TIP=(CATEGORY RESPONSE)

NONE

suppresses data tips and URLs (if requested) from the plot.

**Tip** The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

**TIPFORMAT=(role-format-list)**

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

*(role-format-list) a space-separated list of role-name = format pairs.*
### TIP= (RESPONSE)

**Example**

```plaintext
TIP=(RESPONSE)
TIPFORMAT=(RESPONSE=DOLLAR12.)
```

**Default**
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles.

### TIPLABEL= (role-label-list)

**specifies display labels for tip columns.** This option provides a way to control the labels of columns that appear in data tips.

**role-label-list**
a space-separated list of role-name ="string" pairs.

**Example**

```plaintext
TIP=(RESPONSE)
TIPLABEL=(RESPONSE="Average Sales")
```

**Default**
The column label or column name of the column assigned to the role.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles.

### URL= string-column

**specifies an HTML page to display when a pie slice is selected.**

**string-column**
specifies a column that contains a valid HTML page reference (HREF) for each pie slice that is to have an active link.

**Example**

```plaintext
http://www.sas.com
```

**Restriction**
A generated Other slice does not have a URL. See OTHERSLICE=.

**Requirement**
To generate a plot with selectable pie slices, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interaction**
This option has no effect when TIP=NONE.

**Tips**
The URL value can be blank for some pie slices, meaning that no action is taken when the corresponding slice is selected.

The URL value can be the same for any CATEGORY and RESPONSE pairs. In that case, the same action is taken when the pie slices for those pairs are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to
Optional Response Argument

**RESPONSE=** *numeric-column | expression*

specifies response values that are read from a numeric column or an expression.

Details

The input data for the PIECHART statement is raw, unsummarized input data. The PIECHART performs discrete binning for the Category column and calculates appropriate summarization statistics (sum, mean, and so on) based on the setting for the **STAT=** option.

The fill color of each pie slice is derived from the Color attribute of the GraphData1–GraphDataN style elements as described in "Attribute Rotation Patterns" in SAS Graph Template Language: User’s Guide. The default order of the pie slices depends on the data type of the CATEGORY values:

- For numeric data, the slices appear in the ascending order of the unformatted data values.
- For discrete data, the slices appear in data order.

You can use the **START=** and **CATEGORYDIRECTION=** options to control the pie slice positions and display order.

By default, the pie slices are labeled with the CATEGORY and RESPONSE values, which are displayed inside the slices. You can use the **DATALABELLOCATION=** and **DATALABELCONTENT=** options to control where the pie slices are labeled and the label content.

By default, if two or more slices take up less than 4% of a pie, then an “other” slice is created by consolidating those small slices. To change the default criteria, use the **OTHERSLICE=** and **OTHERSLICEOPTS=** options. The calculated “other” slice is displayed as the last slice in the pie, and as the last legend entry for the pie. If a category value is the same as the “other” slice label, then two slices might be displayed with the same label ("Other" by default) and different fill attributes. In that case, both slices are represented in the pie legend.

To create a pie slice for missing CATEGORY values, specify **INCLUDEMISSINGDISCRETE=TRUE** in the BEGINGRAPH statement. The fill color of the missing category slice is assigned the fill color from the GraphMissing style element except when a user-defined format is applied to the category value. In that case, the missing category slice is assigned the fill color from a GraphData1–GraphDataN style element in data order instead.

---

**Note:** The PIECHART statement does not honor the **MISSING=** system option. Regardless of the **MISSING=** system option value, unless a user-defined format is applied to the value, the default missing-numeric-value character (.) is used to depict missing numeric values.

---

Example: PIECHART Statement

The following graph was generated by the "Example Program" on page 699:
Example Program

```
proc template;
define statgraph simplepie;
begingraph;
    entrytitle "Car Models by Origin";
    layout region;
        piechart category=origin / datalabellocation=outside;
    endlayout;
endgraph;
end;
run;
proc sgrender data=sashelp.cars
    template=simplepie;
run;
```

POLYGONPLOT Statement

Draws a polygon from data that is stored in a data set.

Restriction: The POLYGONPLOT statement does not support CAS in-memory tables. If the polygon plot data is stored in a CAS table, the polygon plot is not drawn.

Note: This statement is valid starting in SAS 9.4M1.

Tip: Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35.
information about the ODS GRAPHICS statement SUBPIXEL= option, see "ODS GRAPHICS Statement" on page 1483.

Syntax
POLYGONPLOT X=column | expression Y=column | expression
ID=column | expression < /options(s)>

Summary of Optional Arguments

Appearance options
- **ANTIALIAS=**AUTO | OFF
  specifies whether anti-aliasing is turned off for this plot.
- **BACKLIGHT=**number | AUTO
  specifies a back-light effect for the polygon label text.
- **COLORMODEL=**color-ramp-style-element | (color-list)
  specifies a color ramp to use with the COLORRESPONSE= option.
- **COLORRESPONSE=**numeric-column | range-attr-var | expression
  specifies the numeric column or range attribute map variable to use to determine the polygon colors.
- **DATASKIN=**NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of filled polygons.
- **DATATRANSPARENCY=**number
  specifies the degree of the transparency of the polygon fill, outline, and label, when these attributes are displayed.
- **DISPLAY=** (display-options) | STANDARD | ALL
  specifies which polygon features to display.
- **FILLATTRS=**style-element | style-element (fill-options) | (fill-options)
  specifies the appearance of the filled polygon areas.
- **FILLPATTERNATTRS=**style-element | (fill-pattern-options)
  specifies the appearance of the pattern-filled areas.
- **OUTLINEATTRS=**style-element | style-element (line-options) | (line-options)
  specifies the appearance of the polygon outline.
- **REVERSECOLORMODEL=**TRUE | FALSE
  specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.
- **ROTATE=**numeric-column | numeric-constant | expression
  specifies the angle of rotation for the polygon, measured in degrees.
- **ROTATELABEL=**AUTO | NONE | VERTICAL
  specifies the rotation of the polygon label with respect to the rotation of the polygon.

Axes options
- **PRIMARY=**TRUE | FALSE
  specifies that the data columns for this plot and the plot type be used for determining default axis features.
- **XAXIS=**X | X2
  specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.
- **YAXIS=**Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

 ROLENAME=(role-name-list)
 specifies user-defined roles that can be used to display information in the data tips.

 TIP=(role-list) | NONE
 specifies the information to display when the cursor is positioned over the polygon.

 TIPFORMAT=(role-format-list)
 specifies display formats for tip columns.

 TIPLABEL=(role-label-list)
 specifies display labels for tip columns.

Label options

 LABEL=column | expression
 specifies the label for the polygon.

 LABELATTRS=style-element | style-element (text-options) | (text-options)
 specifies the color and font attributes of the polygon label.

 LABELLOCATION=INSIDEBOX | OUTSIDEBOX | OUTSIDE
 specifies the location of the polygon label.

 LABELPOSITION= CENTER | XMIN | XMAX | YMIN | YMAX
 specifies the position of the polygon label with respect to the label location.

 LABELSPLIT=TRUE | FALSE
 specifies whether to split the polygon label at the specified split characters.

 LABELSPLITCHAR="character-list"
 specifies one or more characters on which the polygon label can be split if needed.

 LABELSPLITCHARDROP=TRUE | FALSE
 specifies whether the split characters are included in the polygon label.

 LABELSPLITJUSTIFY= AUTO | CENTER | LEFT | RIGHT
 specifies the justification of the strings that are inside the polygon label blocks.

 LEGENDLABEL="string"
 specifies a label to be used in a discrete legend for this plot.

Midpoint options

 GROUP=column | discrete-attr-var | expression
 creates a separate and visually distinctive polygon for each unique grouping value.

 INCLUDEMISSINGGROUP=TRUE | FALSE
 specifies whether missing values of the group variable are included in the plot.

 XOFFSET=numeric-column | expression
 specifies an individual offset from the discrete X value on each vertex of the polygon.

 YOFFSET=numeric-column | expression
 specifies an individual offset from the discrete Y value on each vertex of the polygon.
ODS options

\texttt{URL=string-column}

specifies an HTML page that is displayed when the polygon is selected.

Plot reference options

\texttt{NAME=\"string\"}

assigns a name to this plot statement for reference in other template statements.

Required Arguments

\texttt{X=column \mid expression}

specifies the column for the X values.

Note: A missing value in the X column signals the start of the data for a hole in the polygon. See “Drawing Holes in a Polygon” on page 723.

\texttt{Y=column \mid expression}

specifies the column for the Y values.

Note: A missing value in the Y column signals the start of the data for a hole in the polygon. See “Drawing Holes in a Polygon” on page 723.

\texttt{ID=column \mid expression}

specifies the column that contains the ID value that is associated with each polygon.

Restriction: Only unformatted values in the ID= column are used.

Requirements: All of the observations for a single polygon must have the same ID value.

When multiple polygons are defined in the same data set, all of the observations for a given ID must be defined contiguously. Interspersing the ID observations in the data set can produce unexpected results.

Note: Observations that have a missing value in the ID column are ignored by the POLYGONPLOT statement.

Optional Arguments

\texttt{ANTIALIAS=AUTO \mid OFF}

specifies whether anti-aliasing is turned off for this plot.

\hspace{1em} \textbf{Note}: This feature applies to \texttt{SAS 9.4M2} and to later releases.

\hspace{1em} AUTO

specifies that anti-aliasing is controlled by the \texttt{ANTIALIAS=} option in the ODS GRAPHICS statement.

\hspace{1em} OFF

specifies that anti-aliasing is always disabled for this plot.

Default: AUTO
This option overrides the ANTIALIAS= option in the ODS GRAPHICS statement.

**BACKLIGHT=number | AUTO**
specifies a back-light effect for the polygon label text.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

The effect is applied only to the polygon label text.

*number*
  specifies the degree of the back-light effect.
  
  **Range** 0–1, where 0 specifies no effect and 1 specifies maximum effect

*AUTO*
  the system selects an appropriate level for the back-light effect.
  
  If the GROUP= or COLORRESPONSE= option is in effect, BACKLIGHT=0.75. Otherwise, BACKLIGHT=0.5.

The following figure shows the effect on a polygon label located inside the polygon bounding box.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>AUTO</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="light-text-no-backlight" alt="Label" /></td>
<td><img src="light-text-backlight" alt="Label" /></td>
<td><img src="dark-text-backlight" alt="Label" /></td>
</tr>
</tbody>
</table>

The back light is based on text color. For dark colors, a contrasting white back-light effect is used. For lighter colors, a contrasting black back-light effect is used. The following figure shows the two back-light types when BACKLIGHT=1.

<table>
<thead>
<tr>
<th>Black Text</th>
<th>Light-Gray Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="dark-text-no-backlight" alt="Label" /></td>
<td><img src="light-text-no-backlight" alt="Label" /></td>
</tr>
</tbody>
</table>

**Default**
  0 (no back-light effect)

**Restriction**
  Vector graphics output cannot be generated when the back-light effect is applied. If you request vector graphics output and enable the back-light effect, an image is generated instead.

**Interaction**
  The **LABEL**= option must be specified for this option to have any effect.

**Note**
  If the background color is white, the white backlight effect for dark text colors is not visible. Conversely, if the background color is black, the black backlight effect for light text colors is not visible.

**Tip**
  The BACKLIGHT= option is most effective when the text color has a low level of contrast with the background or when the background is cluttered.
**COLORMODEL** = *color-ramp-style-element | (color-list)*
specifies a color ramp to use with the COLORRESPONSE= option.

*color-ramp-style-element*
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

**STARTCOLOR**
specifies the color for the smallest data value of the COLORRESPONSE= column.

**NEUTRALCOLOR**
specifies the color for the midpoint of the range of the COLORRESPONSE= column.

**ENDCOLOR**
specifies the color for the highest data value of the COLORRESPONSE= column.

*(color-list)*
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement** The list of colors must be enclosed in parentheses.

**See** “color” on page 1480

**Defaults**
- For a filled polygon, the ThreeColorRamp style element
- For an unfilled polygon, the ThreeColorAltRamp style element

**Interaction**
For this option to take effect, the COLORRESPONSE= option must also be specified.

**Tip** To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

**COLORRESPONSE** = *numeric-column | range-attr-var | expression*
specifies the numeric column or range attribute map variable to use to determine the polygon colors.

*range-attr-var*
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“RANGEATTRVAR Statement”

**Restriction** A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

**See** “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.
The COLORRESPONSE= value should remain constant for the same ID value. Otherwise, unexpected results might occur.

Prior to SAS 9.4M5, when the GROUP= option is specified with the COLORRESPONSE= option, the GROUP= option is ignored. Starting with SAS 9.4M5, the COLORRESPONSE= option controls the color of the group fill patterns when the fill patterns are displayed.

When fill is displayed, this option overrides suboption COLOR= in the FILLATTRS= option and varies the fill color according to the color gradient or the attribute map.

When only the outline is displayed, this option overrides suboption COLOR= in the OUTLINEATTRS= option and varies the outline color according to the color gradient or the attribute map.

To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**

enhances the visual appearance of filled polygons. The following figure shows a polygon with each of the skins applied.

![Polygon skins](image)

The DATASKIN= option value that is specified in the BEGINGRAPH statement. If not specified, then the GraphSkins:DataSkin style element value is used.

Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

For this option to have any effect, the DISPLAY= option must include FILL.

This option overrides the BEGINGRAPH statement DATASKIN= option.
The appearance of the data skin is based on the `FILLATTRS=` color.

When a data skin is applied, all polygon outlines are set by the skin and the `OUTLINEATTRS=` option is ignored.

**DATATRANSPARENCY=number**  
specifies the degree of the transparency of the polygon fill, outline, and label, when these attributes are displayed.

- **Default**: 0  
- **Range**: 0–1, where 0 is opaque and 1 is entirely transparent  
- **Tip**: You can use the `FILLATTRS=` option to set transparency for just the filled polygon areas. You can combine this option with `FILLATTRS=` to set one transparency for the polygon outline and label and a different transparency for the polygon fill. Example:  
  
  ```  
  datatransparency=0.2 fillattrs=(transparency=0.6)  
  ```

**DISPLAY=(display-options) | STANDARD | ALL**  
specifies which polygon features to display.

- **(display-options)**  
specifies a space-separated list of one or more display options enclosed in parentheses.

  The list can include one or more of the following options:

  - **FILL**: displays the polygon fill
  - **FILLPATTERN**: displays a pattern-filled polygon
  - **OUTLINE**: displays the polygon outline
  - **STANDARD**: specifies OUTLINE only
  - **ALL**: specifies all features: FILL, FILLPATTERN, and OUTLINE

- **Note**: This feature applies to SAS 9.4M5 and to later releases.

- **Tip**: By default, this option has no effect if the active ODS style does not specify a fill pattern. To display a fill pattern in that case, select an ODS style that supports fill patterns or use the `FILLPATTERNATTRS=` option to specify a fill pattern for the polygon.

  ```  
  OUTLINEATTRS= and FILLATTRS= options to control the appearance of the polygon outline and fill color. Starting with SAS 9.4M5, the GraphPolygon:DisplayOpts style reference, if defined in the active ODS style. Otherwise, STANDARD.  
  ```
9.4M5, use the `FILLPATTERNATTRS=` on page 707 option to control the appearance of the polygon fill pattern.

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the filled polygon areas.

**Defaults**
- For non-grouped data, the GraphDataDefault:Color style reference.
- For grouped data, the Color attribute of GraphData1–GraphDataN style elements.

**Interactions**
- For this option to have any effect, the fill must be enabled by the ODS style or by the `DISPLAY=` option.
- When `COLORRESPONSE=` is in effect and the `DISPLAY=` option enables FILL display, the FILLATTRS= suboption COLOR= is ignored, and the polygon fill colors vary according to the gradient.

**Tip**
The `DATATRANSPARENCY=` option sets the transparency for the polygon fill and outline. You can combine this option with `DATATRANSPARENCY=` to set one transparency for the outlines but a different transparency for the fills. Example:
```
datatransparency=0.2 fillattrs=(transparency=0.6)
```

**See**
- “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.
- “Fill Color Options” on page 1520 for available `fill-options`.

**FILLPATTERNATTRS=style-element | (fill-pattern-options)**
specifies the appearance of the pattern-filled areas.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Defaults**
- For non-grouped data, the ContrastColor and FillPattern attributes of GraphDataDefault.
- For grouped data, the ContrastColor and FillPattern attributes of GraphData1–GraphDataN.

**Interactions**
- For this option to take effect, the `DISPLAY=` option must include FILLPATTERN among the display options.
- Starting with SAS 9.4M5, when `COLORRESPONSE=` is in effect and the `DISPLAY=` option enables FILLPATTERN display, the FILLPATTERNATTRS= suboption COLOR= is ignored, and the fill-pattern colors vary according to the gradient.

**Tip**
In order to change the fill pattern for each group value in ODS styles such as HTMLBlue that use color-priority attribute rotation, specify ATTRPRIORITY=NONE in an ODS GRAPHICS statement or in the BEGINGRAPH statement in your graph template. For more information, see “Attribute Rotation Patterns” in SAS Graph Template Language: User’s Guide.
GROUP=column | discrete-attr-var | expression
creates a separate and visually distinctive polygon for each unique grouping value.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

A distinct polygon is created for each group value by varying the visual attributes of the polygon display features. The display features are controlled by the current ODS style or by the DISPLAY= option. The default group appearance for each display feature is shown in the following table.

<table>
<thead>
<tr>
<th>Display Feature</th>
<th>Style Attributes That Control Default Group Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill color</td>
<td>Color attribute of a GraphData1–GraphDataN style element or the GraphMissing style element.</td>
</tr>
<tr>
<td>Fill pattern</td>
<td>FillPattern and ContrastColor attributes of a GraphData1–GraphDataN style element or the GraphMissing style element. The ContrastColor attribute controls the color of the fill pattern.</td>
</tr>
<tr>
<td>Outline</td>
<td>if the outline is enabled with fill color, fill pattern, or both, ContrastColor attribute of a GraphData1–GraphDataN style element or the GraphMissing style element with a solid line style</td>
</tr>
<tr>
<td></td>
<td>if the outline is the only display feature enabled, ContrastColor and LineStyle attributes of a GraphData1–GraphDataN style element or the GraphMissing style element</td>
</tr>
</tbody>
</table>

1 Fill color and the outline are displayed by default.
2 Fill patterns are valid for polygon plots starting with SAS 9.4M5.

Requirement The group value must remain constant for the same ID value. Otherwise, the results are unpredictable.

Interactions This option is ignored if the COLORRESPONSE= option is also used.
If a discrete attribute map variable is specified, then the colors and outline patterns are mapped according to the associated DISCRETEATTRMAP statement. See "DISCRETEATTRMAP Statement" on page 1425. Otherwise, the colors and outline patterns are mapped according to data order.

The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Note The group values should remain constant for the same ID value.

See "DISCRETEATTRVAR Statement" on page 1435

**INCLUDEMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the group variable are included in the plot.

Default TRUE

Restriction This option is ignored and missing group values are not displayed when ID= and GROUP= specify the same values.

Interaction For this option to take effect, the GROUP= option must also be specified.

Tip The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See “boolean ” on page 1479 for other Boolean values that you can use.

**LABEL=column | expression**

specifies the label for the polygon.

Default No label is displayed for the polygon

Note The label text should be the same for all of the observations for a polygon ID. When different labels are specified for the same ID, the label that is specified in the first observation for that ID is used.

Tips The default label text color is based on the use of polygon fill and outline colors, and on whether the GROUP= or COLORRESPONSE= option is specified. To change the label text color and font, use the LABELATTRS= option.

Use the LABELLOCATION= and LABELPOSITION= options to change the location of the polygon label.

For long labels, use the LABELSPLIT=, LABELSPLITCHAR=, LABELSPLITCHARTDROP=, and LABELSPLITJUSTIFY= options to split the label into multiple lines.

**LABELATTRS=style-element | style-element (text-options) | (text-options)**

specifies the color and font attributes of the polygon label.
For non-grouped data, the GraphDataText style element, unless COLORRESPONSE= is specified. When COLORRESPONSE= is specified, the label color is determined by the outline color when outlines are displayed, or by the GraphDataText style element color attribute when outlines are not displayed. All other text attributes are derived from the GraphDataText style element.

For grouped data, the GraphData1–GraphDataN style elements. The label color is determined by the contrast color attribute.

If one or more text options are specified and they do not include all of the font properties (such as color, family, size, weight, and style), then the non-specified properties are derived from the GraphLabelText style element or from a GraphData1–GraphDataN style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

LABELLOCATION=INSIDEBOX | OUTSIDEBOX | OUTSIDE
specifies the location of the polygon label.

INSIDEBOX
locates the label inside the bounding box of the polygon.

OUTSIDEBOX
locates the label outside the bounding box of the polygon but inside the plot area.

OUTSIDE
locates the label outside the plot area.

Default INSIDEBOX

Restriction OUTSIDE cannot be used when the POLYGONPLOT statement is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE in which the axes might be external to the grid.

Interaction The LABEL= option must be specified for this option to have any effect.

Tip The label's exact position is relative to the polygon’s X and Y data ranges and is determined by the combination of this option and the LABELPOSITION= option.

LABELPOSITION=CENTER | XMIN | XMAX | YMIN | YMAX
specifies the position of the polygon label with respect to the label location.

CENTER
centers the label in the polygon's bounding box

XMIN
positions the label at the polygon's minimum X value and centers it in the Y-value range

XMAX
positions the label at the maximum X value and centers it in the Y-value range
YMIN
positions the label at the minimum Y value and centers it in the X-value range

YMAX
positions the label at the maximum Y value and centers it in the X-value range

The following figure shows the label positions for each of the label locations that are specified by the LABELLOCATION= option.

**Defaults**
- CENTER when LABELLOCATION=INSIDEbbox
- YMAX when LABELLOCATION=OUTSIDEbbox or LABELLOCATION=OUTSIDE

**Restriction**
- CENTER is valid only when LABELLOCATION=INSIDEbbox.

**Interaction**
- When LABELLOCATION=OUTSIDE, increasing label length might cause the available plot area to decrease.

**Tip**
- When LABELLOCATION=OUTSIDE, the polygon label might collide with the axis tick values on the orthogonal axis. In that case, if the secondary orthogonal axis is not being used, specify the opposite end of the axis. Otherwise, change LABELLOCATION= to INSIDEbbox or OUTSIDEbbox.

**LABELSPLIT=TRUE | FALSE**
- specifies whether to split the polygon label at the specified split characters.
- When this option is set to TRUE, the polygon label is split unconditionally at each occurrence of any of the specified split characters.
FALSE. The polygon label is not split.

The LABEL= option must also be specified.

The LABELSPLITCHAR= option specifies one or more characters on which splits can occur.

"boolean " on page 1479 for other Boolean values that you can use.

LABELSPLITCHAR="character-list"

specifies one or more characters on which the polygon label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the polygon label. In that case, all of the specified split characters together are treated as a single split character.

When LABEL= is specified and LABELSPLIT=TRUE, the polygon label is split unconditionally at each occurrence of any of the specified split characters. If the polygon label does not contain any of the specified characters, then the label is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

A blank space

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

labelsplitchar="abc"

The LABEL= option and the LABELSPLIT=TRUE option must also be specified.

The LABELSPLITCHARDROP= option specifies whether the split characters are included in the polygon label or are dropped.

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Use the LABELSPLITJUSTIFY= option to specify the justification of the strings in the polygon label block.

LABELSPLITCHARDROP=TRUE | FALSE

specifies whether the split characters are included in the polygon label.

TRUE

drops the split characters from the polygon label.

FALSE

includes the split characters in the polygon label.

When LABELSPLIT=TRUE and LABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The
characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of polygon label Product*Group*A split on * when LABELSPLITCHARDROP=TRUE and LABELSPLITCHARDROP=FALSE.

<table>
<thead>
<tr>
<th>LABELSPLITCHARDROP=TRUE</th>
<th>Product<em>Group</em>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>LABELSPLITCHARDROP=FALSE</td>
<td>Product<em>Group</em>A</td>
</tr>
</tbody>
</table>

Default: TRUE. The split characters are dropped from the polygon label.

Requirement: The LABEL= option and the LABELSPLIT=TRUE option must also be specified.

Interaction: The LABELSPLITCHAR= option specifies the split characters.

See “boolean” on page 1479 for other Boolean values that you can use.

**LABELSPLITJUSTIFY=**AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the polygon label blocks.

**AUTO**

justifies the labels based on the LABELPOSITION= option.

<table>
<thead>
<tr>
<th>LABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMIN</td>
<td>RIGHT</td>
</tr>
<tr>
<td>XMAX</td>
<td>LEFT</td>
</tr>
<tr>
<td>CENTER, YMAX, or YMIN</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

**CENTER**

**LEFT**

**RIGHT**

justifies the labels center, left, or right, as specified.

The following figure shows an example in which LABELPOSITION=YMAX.

Note: The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.
In this case, because LABELPOSITION=YMAX, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored.

**Default**  AUTO  

**Requirement**  The LABEL= option and the LABELSPLIT=TRUE option must also be specified.

**LEGENDLABEL=**"string"

specifies a label to be used in a discrete legend for this plot.

**Default**  The string specified on the NAME= option.

**Restriction**  This option applies only to an associated DISCRETELEGEND statement.

**Interaction**  If the GROUP= option is specified, then this option is ignored.

**NAME=**"string"

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**  The string is used as the default legend label if the LEGENDLABEL= option is not used.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the appearance of the polygon outline.

**Defaults**  For non-grouped data, the GraphOutlines style element.

For grouped data, unfilled polygons use both the CONTRASTCOLOR and PATTERN attributes of the GraphData1–GraphDataN style elements. Filled polygons use only the CONTRASTCOLOR attribute.

**Interactions**  For this option to have any effect, outlines must be enabled by the ODS style or by the DISPLAY= option.

If the DATASKIN= option applies a data skin, then this option is ignored.

When the COLORRESPONSE= and DISPLAY=(OUTLINE) options are in effect, the OUTLINEATTRS= suboption COLOR= is ignored, and the polygon outline colors vary according to the gradient.

**See**  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.
PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for
determining default axis features. This option is needed only when two or more
plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDED or
LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the
primary plot on a per-axis basis. When no plot is designated as the
primary plot, the first plot that can be a primary plot is considered the
primary plot. If multiple plots specify PRIMARY=TRUE for the same
axis, then the last such plot encountered is considered the primary
plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

REVERSECOLORMODEL=TRUE | FALSE
specifies whether to reverse the gradient (color ramp) that is defined by either
the ODS style that is in effect or by the COLORMODEL= option.

Default FALSE

See COLORMODEL=

“boolean ” on page 1479 for other Boolean values that you can use.

ROLENAME=(role-name-list)
specifies user-defined roles that can be used to display information in the data
tips. This option provides a way to add to the data columns that appear in data
tips that are specified by the TIP= option.

(role-name-list)
a space-separated list of role-name = column pairs.

Example The following example assigns the column Obs to the user-defined
role TIP:
ROLENAME=(TIP1=OBS)

Default No user-defined roles

Requirement The role names that you choose must be unique and different from
the predefined roles ID, COLORRESPONSE, GROUP, LABEL, and
URL.

ROTATE=numeric-column | numeric-constant | expression
specifies the angle of rotation for the polygon, measured in degrees. Positive
angles rotate the image counter clockwise, and negative angles rotate the image
clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0. No rotation is performed.
Interaction

When this option is specified, the LABELLOCATION= and LABELPOSITION= options are overridden with LABELLOCATION=INSIDEBOX and LABELPOSITION=CENTER.

Notes

The rotation angle is measured in screen coordinates.
A missing value in the rotation data is treated as 0.
Rotating a polygon does not change data ranges that are reported to the axes. As a result, clipping might occur in some cases.

**ROTATELABEL=** AUTO | NONE | VERTICAL

specifies the rotation of the polygon label with respect to the rotation of the polygon.

**AUTO**

rotates the label with the rotation of the polygon.

**NONE**

does not rotate the label with the rotation of the polygon.

The label position remains fixed regardless of the polygon rotation.

**VERTICAL**

rotates the label to a vertical position.

**Restriction**

VERTICAL is valid only when the polygon is not rotated (ROTATE=0). When the polygon is rotated, ROTATELABEL=VERTICAL is ignored, and the default (AUTO) is used instead.

**Default**

AUTO

**Interaction**

The **LABEL**= option must be specified for this option to have any effect.

**TIP=(role-list) | NONE**

specifies the information to display when the cursor is positioned over the polygon. If you use this option, it replaces all of the information that is displayed by default. You can specify roles for columns that do not contribute to the polygon plot along with roles that do.

**(role-list)**

an ordered, space-separated list of unique POLYGONPLOT and user-defined roles.

POLYGONPLOT roles include ID, COLORRESPONSE, GROUP, LABEL, and URL.

**Tip**

User-defined roles are defined with the **ROLENAMESPACE**= option.

**Example**

The following example displays the columns that are assigned to the roles ID and URL, and the columns XOffset and YOffset in the data tips. The XOffset and YOffset columns are not assigned to any predefined POLYGONPLOT role, so they must first be assigned a role:

ROLENAMESPACE=(TIP1=XOFFSET TIP2=YOFFSET)
TIP=(ID TIP1 TIP2 URL)
NONE
suppresses data tips and URLs (if requested) from the plot.

Default The columns that are assigned to the following roles are automatically included in the data tip information: ID, COLORRESPONSE or GROUP, LABEL, and URL.

Requirement To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction Only the roles that appear in the TIP= option are used.

Requirement A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

Default The column label or column name of the column assigned to the role.

Restriction Only the roles that appear in the TIP= option are used.

URL=string-column
specifies an HTML page that is displayed when the polygon is selected.
**string-column**
specifies a column that contains a valid HTML page reference (HREF) for each polygon that is to have an active link.

**Example**
http://www.sas.com

**Requirement**
To generate a plot with selectable polygons, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interactions**
This option has no effect when **TIP=NONE**.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Note**
The URL values should remain constant for the same ID value.

**Tip**
By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See **DRILLTARGET=** on page 1489.

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default**
X

**Interaction**
The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

**XOFFSET=**numeric-column | expression
specifies an individual offset from the discrete X value on each vertex of the polygon.

**Default**
0 (all polygon vertices are centered on the discrete X values)

**Range**
-0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is to the right on discrete X values. If option REVERSE=TRUE is specified in the layout's X-axis options, then the offset direction is also reversed.

**Restriction**
This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

**Interaction**
This option overrides the DISCRETEOFFSET= option.

**Tip**
Setting the discrete offset for the plots does not affect the axis minimum offset and maximum offset. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to
increase the axis minimum and maximum offsets in order to accommodate the discrete offset.

**YAXIS=** `Y | Y2`
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default `Y`

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YOFFSET=** `numeric-column | expression`
specifies an individual offset from the discrete Y value on each vertex of the polygon.

Default 0 (all polygon vertices are centered on the discrete Y values)

Range –0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is up on discrete Y values. If option `REVERSE=TRUE` is specified in the layout's Y-axis options, then the offset direction is also reversed.

Restriction This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

Interaction This option overrides the `DISCRETEOFFSET=` option.

Tip Setting the discrete offset for the plots does not affect the axis minimum offset and maximum offset. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the `OFFSETMIN=` and `OFFSETMAX=` axis options to increase the axis minimum and maximum offsets in order to accommodate the discrete offset.

**Details**

**Overview**
The POLYGONPLOT statement draws a polygon from a series of X-Y value pairs that are stored in a SAS data set. The first X-Y value pair defines the starting point of the polygon. The next X-Y pair in the data specifies the coordinates of the first vertex. A line segment is drawn from the starting point to the first vertex. For subsequent X-Y pairs, a line segment is drawn from the previous vertex to the current vertex. This pattern repeats until all of the segments have been drawn. If the last segment does not close the polygon, then the POLYGONPLOT statement automatically draws a segment from the last vertex back to the starting point in order to close the polygon.

Using the POLYGONPLOT statement, you can draw any data-driven shape on your graph, which enables you to highlight data features, outline data boundaries, and so on. Unlike DRAW statements in a BEGINPOLYGON block, the following conditions apply to the POLYGONPLOT statement:

- you need to modify only the polygon data in the graph data set to modify the polygon. You do not need to modify the template code. (See “BEGINPOLYGON Statement” on page 1329.)
you can draw polygons between plots. The BEGINPOLYGON block and DRAW statements can draw polygons only on top of or behind the graph.

Requirements for the Polygon Data Set

In the simplest case of a single polygon, your data set must provide an X, Y, and ID column that stores the X-Y values and the ID for your polygon. The X-Y values in the first data-set observation must specify the starting point of your polygon. The X-Y values in the subsequent observations must provide the coordinates of each vertex in the order in which the polygon is to be drawn. There should be no gaps in the data. If your last observation does not close the polygon, then the POLYGONPLOT statement automatically draws a segment from your last vertex back to the starting point in order to close the polygon.

If you want to draw multiple polygons, then your ID column must specify a unique identifier value for each polygon. The identifier value associates the observations in the data set with a specific polygon. All of the observations for each individual polygon must be grouped together by ID and must be arranged in the order in which the polygon segments are to be drawn.

The following table shows additional columns that you can use to customize your polygon plot.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLORRESPONSE (p. 704)</td>
<td>Numeric</td>
<td>Specifies the numeric column or range attribute map variable that is used to determine the polygon colors.</td>
</tr>
<tr>
<td>GROUP (p. 708)</td>
<td>Numeric or character</td>
<td>Creates a separate polygon color, outline pattern, or both for each unique grouping that is specified.</td>
</tr>
<tr>
<td>LABEL (p. 709)</td>
<td>Numeric or character</td>
<td>Specifies the label for the polygon.</td>
</tr>
<tr>
<td>URL (p. 717)</td>
<td>Character</td>
<td>Specifies an HTML page that is displayed when the polygon is selected.</td>
</tr>
<tr>
<td>XOFFSET (p. 718)</td>
<td>Numeric</td>
<td>Specifies an individual offset from the discrete X value on each vertex of the polygon.</td>
</tr>
<tr>
<td>YOFFSET (p. 719)</td>
<td>Numeric</td>
<td>Specifies an individual offset from the discrete Y value on each vertex of the polygon.</td>
</tr>
</tbody>
</table>

1 You can specify any valid column name for these columns in your data set.

Drawing a Single Polygon

For a single polygon, the polygon data set contains an X column and Y column that define the polygon vertices, and an ID column that specifies a constant value. The polygon segments are drawn in the order in which they occur in the data. If the polygon overlaps any graphics elements that were drawn earlier, those elements are
obscured. In that case, you can use transparency to enable the underlying graphics elements to show through.

Here is example data for a simple four-sided polygon that is identified as P1 and that starts at point X=40, Y=100.

<table>
<thead>
<tr>
<th>Obs</th>
<th>id</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>P1</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>P1</td>
<td>20</td>
<td>220</td>
</tr>
<tr>
<td>3</td>
<td>P1</td>
<td>160</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>P1</td>
<td>180</td>
<td>80</td>
</tr>
<tr>
<td>5</td>
<td>P1</td>
<td>40</td>
<td>100</td>
</tr>
</tbody>
</table>

The following figure shows how the polygon is drawn. Grid lines are provided to help you locate the polygon vertices in the output.

The polygon starting point is X=40, Y=100 (shown in red). From the starting point, the segments are drawn in data order. Data order is in a clockwise direction, as indicated by the gray arrow. Although the last observation (X=40, Y=100) is provided in this example, it is not required. If the last observation is not provided in the data, then the POLYGONPLOT statement draws the last segment automatically in order to close the polygon.

For an example, see “Example 1: Drawing a Simple Polygon That Highlights Data” on page 724.

Drawing Multiple Polygons

For multiple polygons, the POLYGONPLOT data ID column specifies a unique identifier for all of the observations that are associated with each polygon. The X and Y columns specify the polygon vertices. The polygons are overlaid on the graph in the order in which they occur in the data. For overlapping polygons, each polygon obscures part or all of the polygons and graphics elements that were drawn before it. In that case, you can use transparency to enable the underlying polygons and graphics elements to show through.
Here is example data for three separate polygons.

### Polygon Data

<table>
<thead>
<tr>
<th>Obs</th>
<th>id</th>
<th>x</th>
<th>y</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>ID=1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>20</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>20</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>30</td>
<td>0</td>
<td>ID=2</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>40</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>30</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>60</td>
<td>0</td>
<td>ID=3</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>80</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>80</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>70</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>60</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>60</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

In addition to the ID column, X column, and Y column, the Label column is added to label the polygons in the output. Notice that the observations for each ID value are grouped together in the data set. The observations for each ID must occur contiguously in the data. Otherwise, unexpected results might occur.

The following figure shows how the polygons are drawn from this data. Grid lines are provided to help you locate the polygon vertices in the output.

![Polygon Diagram](image)

The polygons are drawn in the order in which they appear in the data: rectangle (ID=1), triangle (ID=2), and polygon (ID=3). The red dot on each shape indicates the starting point for that shape. The gray arrow indicates the direction in which the segments are drawn for each shape.
Drawing Holes in a Polygon

Using the POLYGONPLOT statement, you can draw one or more holes inside a polygon. To create data for a polygon with one or more holes:

1 Specify the X and Y values for the outer polygon.

2 To start the data for a hole, add an observation that has missing X and Y values. The missing X and Y values signal the POLYGONPLOT statement that the observations that follow define the data for a hole.

3 Specify the X and Y values for the hole polygon.

4 Repeat Steps 2 and 3 for each additional hole.

Here is example data for a simple polygon that has two holes.

Polygon Data

<table>
<thead>
<tr>
<th>Obs</th>
<th>id</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Observations 1–7 specify the data for the outer polygon. In observation 8, the X and Y values are missing, which indicates the start of the data for the first hole. Observations 9–13 define the data for the first hole polygon. Observation 14 indicates the start of the data for the second hole, which is defined by observations 15–19.

The following figure shows how the polygon is drawn from this data. Grid lines are provided to help you locate the polygon vertices in the output.
The outer polygon is drawn first, starting at point X=1, Y=1. The segments are drawn in data order, which is in a counterclockwise direction as indicated by the gray arrow. The first hole is drawn next, starting at point X=3, Y=8. Its segments are drawn in a clockwise direction. The second hole is drawn last, starting at point X=3, Y=2. Its segments are drawn in a counterclockwise direction.

Examples:

Example 1: Drawing a Simple Polygon That Highlights Data

This example shows you how to use the POLYGONPLOT statement to draw a filled polygon that highlights data in an iris petal dimension scatter plot. The polygon surrounds the markers for the Setosa species in order to highlight the data for that species. This example is a modified version of the example in “Example: BEGINPOLYGON Statement” on page 1335. This version uses the POLYGONPLOT statement instead of a BEGINPOLYGON block so that you can compare the two methods. The following figure shows the output for this example.
Here is the SAS code for this example.

```sas
/* Generate the data for the polygon */
data polydata;
   input polyID polyX polyY label $8-40;
   datalines;
   1  9 2    Setosa
   1 13 5
   1 16 7
   1 17 6
   1 20 5
   1 20 1
   1 17 1
   1 15 0
   1 14 0
   1 11 0
   ;
run;

/* Concatenate the SASHELP.IRIS and polygon data into data set IRIS */
data iris;
   set sashelp.iris polydata;
run;

/* Create the template for the graph */
proc template;
   define statgraph discretelegend;
      begingraph / drawspace=datavalue;
         entrytitle "Iris Petal Dimensions";
         layout overlayequated / equatetype=equate;
         scatterplot x=petallength y=petalwidth / name="s"
            group=species includemissinggroup=false;
         ellipse x=petallength y=petalwidth / type=predicted alpha=.2
      endgraph;
   enddefine;
   title "Iris Petal Dimensions";
   entry "Setosa" / location=left;
   entry "Versicolor" / location=left;
   entry "Virginica" / location=left;
   attribute specialedge / transparency=.5;
   attribute specialedge / transparency=.5;
   attribute specialedge / transparency=.5;
endproc;
```

Fisher's Iris Data
name="p80" legendlabel="80%" outlineattrs=graphconfidence;
ellipse x=petallength y=petalwidth / type=predicted alpha=.05
name="p95" legendlabel="95%" outlineattrs=graphconfidence2;
polygonplot x=polyX y=polyY id=polyID / display=(fill)
  fillattrs=(color=yellow transparency=0.75)
  label=label labellocation=outsidebbox labelposition=ymax;
discretelegend "s" / title="Species:"
  discretelegend "p80" "p95" / across=1 autoalign=(topleft)
  location=inside;
endlayout;
entryfootnote halign=left "Fisher's Iris Data" ;
endgraph;
end;
run;

/* Generate the graph */
proc sgrender data=iris template=discretelegend;
run;

Details

To draw a single polygon, the data set must provide an X column, a Y column, and an ID column. The data specifies the polygon vertices around the Setosa data in a clockwise direction. A Label column is added to provide a label for the polygon in the plot output. Concatenation of the Polydata and Sashelp.Iris data sets results in missing values for the SCATTERPLOT statement grouping variable in the Iris data set. By default, the SCATTERPLOT statement includes missing group values. To exclude the missing group values, the INCLUDEMISSINGGROUP=FALSE option is added to the SCATTERPLOT statement.

In the POLYGONPLOT statement, the DISPLAY= option specifies the polygon fill only. The FILLATTRS= option specifies the fill color as yellow and a fill transparency of 0.75. Rather than using draw statements to draw an annotation for the polygon, this example uses the POLYGONPLOT statement label feature to label the polygon. The LABEL= option specifies the column in the data set that contains the polygon label text. The LABELLOCATION= and LABELPOSITION= options place the polygon label outside of and above the polygon’s bounding box.

To draw the polygon, the POLYGONPLOT statement starts at X=9, Y=2, and draws a segment between each vertex in data order. The last vertex, X=11, Y=0, does not close the polygon. To close the polygon, the POLYGONPLOT statement draws a segment between X=11, Y=0 and X=9, Y=2 automatically.

If you want to highlight the Versicolor data instead of the Setosa data, then you need only modify the data in the Polydata data set to draw a polygon around the Versicolor data instead. You do not have to make any changes to the template code.

Example 2: Drawing a Geographical Boundary

This example shows you how to use the POLYGONPLOT statement to draw an outline of Wake County in North Carolina. The polygon data is extracted from data set Maps.Counties using state ID 37 for North Carolina and county ID 183 for Wake County. The GPROJECT procedure is then used to project the data before it is plotted with the POLYGONPLOT statement. The Maps.Counties data set is installed with SAS/GRAPH. If SAS/GRAPH is not installed at your site, you can download data set Maps.Counties from SAS Maps Online. For more information, see “Map Data Sets, Map Preparation Procedures, and Tools Provided by SAS” in SAS/GRAPH and Base SAS: Mapping Reference.
The following figure shows the output for this example.

Here is the SAS code for this example.

```sas
/* Extract the map data for Wake County, NC. */
data wakecounproj;
   set maps.counties(where=(state=37 and county=183));
   label="Wake County, NC";
run;

/* Project the data */
proc gproject data=wakecounproj out=wakeco dupok;
   id state county;
run;

/* Draw the county border */
proc template;
   define statgraph wakecounty;
      begingraph;
         layout overlay / walldisplay=(fill)
            xaxisopts=(display=none offsetmin=0.05 offsetmax=0.05)
            yaxisopts=(display=none offsetmin=0.05);
         polygonplot x=x y=y id=state / display=(outline fill)
               fillattrs=(color=cxd1d1d1)
               outlineattrs=(color=darkgreen thickness=2px)
               label=label labelattrs=(weight=bold color=darkgreen size=16pt);
      endlayout;
   endgraph;
run;

proc sgrender data=wakeco template=wakecounty;
run;
```
REFERENCELINE Statement

Creates a horizontal or vertical reference line.

Requirement: A REFERENCELINE statement can be used within a 2-D layout (OVERLAY, OVERLAYEQUATED, DATALATTICE, or DATAPANEL) only.

Note: Specifying the X= option creates a line perpendicular to the X axis at an X-intercept. Specifying the Y= option creates a line perpendicular to the Y axis at a Y-intercept.

Syntax

REFERENCELINE X=x-axis-value | column | expression </options>;
REFERENCELINE Y=y-axis-value | column | expression </options>;

Summary of Optional Arguments

Appearance options

CLIP=TRUE | FALSE
specifies whether the reference line data is to be considered when determining the data range for the axis.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the reference lines.

DATATRANSPARENCY=number
specifies the degree of the transparency of the reference line.

DISCRETELINETHICKNESS=number
specifies the line thickness as a fraction of the midpoint spacing.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the reference line.

USEDISCRETELINETHICKNESS=TRUE | FALSE
specifies that the line thickness should be based on a fraction of the midpoint spacing that is set by the DISCRETELINETHICKNESS= option.

Axes options

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Label options

CURVELABEL="string" | column | expression
specifies a label for the reference line or lines.

CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the reference line label(s).

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the reference line label relative to the plot area.

CURVELABELPOSITION=AUTO | MAX | MIN
specifies the position of the reference line label relative to the reference line.

CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the reference line label at the specified split characters.

CURVELABELSPLITCHAR="character-list"
specifies one or more characters on which the reference line label can be split if needed.

CURVELABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the reference line label text.

CURVELABELJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the reference line label block.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options

DISCRETEOFFSET=number
specifies an amount to offset all reference lines from the specified values when the X or Y axis is discrete.

Plot reference options

NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments

You must use either the X= or the Y= argument in the REFERENCELINE statement.

X=x-axis-value | column | expression
specifies the X intercept of the reference line or lines.

Required Arguments

If X is not specified, then Y must be specified.

Values must be the same type as the data type of the X axis. For example, you should use numeric SAS date or time values (or SAS date/time constants) for a time axis.

Unformatted numeric values do not map to a formatted discrete axis. When the X axis is a discrete axis, the X-axis value must be the formatted value that appears on the X axis. If a column is specified for the values in that case, then the specified column must use the same format that is used for the X axis.

Note

When a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

Tips

By default, if the value that is specified for the X= argument is outside of the data range, then the data range is extended to
include the specified intercept. You can change this behavior with the CLIP= option.

You can use the COLN() or COLC() function in an EVAL() expression to specify multiple reference lines on the X axis. See “GTL Functions Used with the EVAL Function” on page 1465.

Y=y-axis-value | column | expression

specifies the Y intercept of the reference line or lines.

Requirements

If Y is not specified, then X must be specified.

Values must be the same type as the data type of the Y axis.

Unformatted numeric values do not map to a formatted discrete axis. When the Y axis is a discrete axis, Y-axis value must be the formatted value that appears on the Y axis. If a column is specified for the values in that case, then the specified column must use the same format that is used for the Y axis.

Note

When a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

Tips

By default, if the value that is specified for the Y= argument is outside of the data range, then the data range is extended to include the specified intercept. You can change this behavior with the CLIP= option.

You can use the COLN() or COLC() function in an EVAL() expression to specify multiple reference lines on the Y axis. See “GTL Functions Used with the EVAL Function” on page 1465.

Optional Arguments

CLIP=TRUE | FALSE

specifies whether the reference line data is to be considered when determining the data range for the axis.

FALSE

specifies that the reference line values are to be considered when the axis range is determined. The reference lines are drawn as follows based on the axis type:

- For a discrete axis, the reference line values that are not already on the axis are added to the end of the axis data list. When applicable, the axis values are then sorted:
  - If the axis values are numeric values, then they are sorted ordinally.
  - If the axis values are character values and a sorting option is applied to the axis, then they are sorted as specified by the sorting option.

Reference lines are then drawn at the specified locations.

- For a linear, log, or time axis, a new axis data list is created by performing a mathematical union of the data values and the reference line values. The reference lines are then drawn at the locations specified.
TRUE

specifies that the reference line values are not to be considered when the axis range is determined. The reference lines are drawn as follows based on the axis type:

- For a discrete axis, if the reference line value exactly matches a value on the axis, a reference line is drawn at that location. Otherwise, the reference line is not drawn.

Note: If the axis values are formatted, then the reference line value must exactly match the formatted axis value in order for the line to be drawn.

- For a linear, log, or time axis, if the reference line value is within the axis data range, then the reference line is drawn at the specified location. Otherwise, the reference line is not drawn.

Default FALSE

See “boolean” on page 1479 for other Boolean values that you can use.

CURVELABEL="string" | column | expression

specifies a label for the reference line or lines.

Requirement If you use the COLN() or COLC() function in the X= or Y= option to specify multiple reference line intercepts, then you must use the COLC() function in the CURVELABEL= option to specify exactly one label for each reference line intercept. Otherwise, this option is ignored. See “GTL Functions Used with the EVAL Function” on page 1465.

Interactions If the X or Y argument specifies a value, then use “string”.

If the X or Y argument specifies a column, then use a column to define the label for each value.

Tip The font and color attributes for the label are specified by the CURVELABELATTRS= option.

CURVELABELATTRS=style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the reference line label(s).

Default The GraphValueText style element.

Interaction For this option to take effect, the CURVELABEL= option must also be used.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

CURVELABELLOCATION=INSIDE | OUTSIDE

specifies the location of the reference line label relative to the plot area.

INSIDE locates the labels inside the plot area

OUTSIDE locates the labels outside the plot area
OUTSIDE cannot be used when the REFERENCELINE is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELPOSITION= option to determine where the line labels appear.

See "Location and Position of Curve Labels" on page 200

CURVELABELPOSITION=AUTO | MAX | MIN

specifies the position of the reference line label relative to the reference line.

AUTO

automatically positions the line label near the line boundary along unused axes whenever possible (typically Y2 and X2) in order to avoid collision with tick values.

MAX

forces the line label to appear near maximum line values (typically, the top or right).

MIN

forces the line label to appear near minimum line values (typically, the bottom or left).

Defaults

AUTO when CURVELABELLOCATION=OUTSIDE.

MAX when CURVELABELLOCATION=INSIDE.

Restriction

The AUTO setting is ignored if CURVELABELLOCATION= INSIDE is specified.

Interactions

For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELLOCATION= option to determine where the line label appears.

Note

When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the reference line label might fall outside of the graph area. In that case, the reference line label might not be displayed or might be positioned incorrectly.

See "Location and Position of Curve Labels" on page 200
CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the reference line label at the specified split characters. When a reference line label is split, the label is split on each occurrence of the specified split characters.

Default FALSE. The reference line label is not split.

Requirement The CURVELABEL= option must also be specified.

Interactions The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.

This option has no effect when CURVELABELPOSITION=AUTO.

See "boolean " on page 1479 for other Boolean values that you can use.

CURVELABELSPLITCHAR="character-list"
specifies one or more characters on which the reference line label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the reference line label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the reference line label is split unconditionally at each occurrence of any of the specified split characters. If the reference line label does not contain any of the specified characters, then the label is not split.

"character-list"
a list of one or more characters with no spaces between them enclosed in quotation marks.

Example To specify the split characters a, b, and c:
curvelabelsplitchar="abc"

Default A blank space.

Requirement The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interactions This option has no effect if CURVELABELPOSITION=AUTO.

The CURVELABELSPLITCHARDROP= option specifies whether the split characters are included in the reference line label or are dropped.

Notes When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip Use the CURVELABELSPLITJUSTIFY= option to specify the justification of the strings in the reference line label block.

CURVELABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the reference line label text.
TRUE  
drops the split characters from the reference line label text.

FALSE  
includes the split characters in the reference line label text. When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a Y-axis reference line label with the following specifications:
- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

Note: The horizontal line to the left of the label represents the maximum end of the reference line for reference.

![Reference Line Label Example]

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default  
TRUE. The split characters are dropped from the reference line label.

Requirement  
The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction  
The CURVELABELSPLITCHAR= option specifies the split characters.

See  
“boolean ” on page 1479 for other Boolean values that you can use.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT  
specifies the justification of the strings that are inside the reference line label block.

AUTO  
justifies the labels based on the CURVELABELPOSITION= option, as shown in the following table.
### CURVELABELPOSITION= Value Justification

<table>
<thead>
<tr>
<th>Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

**CENTER**

**LEFT**

**RIGHT**

justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

---

**Note:** The horizontal line to the left of each label represents the maximum end of the reference line for reference.

---

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

**Default** AUTO

**Requirement** The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction** This option has no effect if CURVELABELPOSITION=AUTO.

**DATASKIN=**

NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the reference lines. The following figure shows a blue reference line with each of the skins applied.
**DATASKIN=** option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

**Restriction**
Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

**Interactions**
This option overrides the BEGINGRAPH statement DATASKIN= option.

The data skin appearance is based on the LINEATTRS= color.

---

**DATATRANSPARENCY=** *number*

specifies the degree of the transparency of the reference line.

**Default** 0

**Range** 0–1, where 0 is opaque and 1 is entirely transparent

**Note** This option does not affect the reference line label.

---

**DISCRETELINETHICKNESS=** *number*

specifies the line thickness as a fraction of the midpoint spacing.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Default** 1 (100%)

**Range** 0–1

**Requirement** The axis must be discrete and usediscretethickness=TRUE must be in effect. Otherwise, this option is ignored.

---

**DISCRETEOFFSET=** *number*

specifies an amount to offset all reference lines from the specified values when the X or Y axis is discrete.

**Default** 0 (no offset, all reference lines are centered on discrete X or Y values)

**Range** -0.5 to +0.5 where 0.5 represents half the distance between discrete ticks. A positive offset is to the right for a vertical reference line and up for a horizontal reference line. If the layout’s axis options set reverse=TRUE, then the offset direction is also reversed.

**Restriction** This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

**Tip** Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the offsetmin= and offsetmax= axis options to increase the...
axis minimum and maximum offsets to accommodate the discrete offset.

**LEGENDLABEL=**"*string*"
 specifies a label to be used in a discrete legend for this plot.

**Default** The *string* specified on the NAME= option.

**Restriction** This option applies only to an associated DISCRETELEGEND statement.

**LINEATTRS=**style-element | style-element (line-options) | (line-options)
 specifies the attributes of the reference line.

**Default** The GraphReference style element.

**Interaction** This option is ignored when USEDISCRETETHICKNESS=TRUE is in effect.

**See** “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element

“.Line Options” on page 1522 for available line-options.

**NAME=**"*string*"
 assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions** The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction** The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**USEDISCRETETHICKNESS=**TRUE | FALSE
 specifies that the line thickness should be based on a fraction of the midpoint spacing that is set by the DISCRETELINETHICKNESS= option.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Default** FALSE

**Requirement** The axis type must be discrete. Otherwise, this option is ignored.

**Interaction** When this option is set to TRUE, the LINEATTRS THICKNESS= option is ignored.

**Tips** Use USEDISCRETETHICKNESS=TRUE with DISCRETELINETHICKNESS=1, for example, to place custom bands behind the midpoint values.

Use the LINEATTRS COLOR= option to specify the color.
Use `DATATRANSPARENCY=` to specify the transparency.

See `DISCRETELINETHICKNESS=` on page 736

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Restriction: Another plot that establishes a data range for the designed axis must be included.

Interactions: This option is ignored if the X= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

Restriction: Another plot that establishes a data range for the designed axis must be included.

Interactions: This option is ignored if the Y= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**Details**
Reference lines are always drawn perpendicular to the axes. They are drawn from one axis boundary to the companion boundary (X to X2 or Y to Y2). Axis offsets do not apply to reference lines.

A `REFERENCELINE` statement can be used only within 2-D overlay-type layouts (OVERLAY, OVERLAYEQUATED, or PROTOTYPE). A stand-alone plot statement that provides a sufficient data range for determining axis extents must be included in the layout. For example, a `REFERENCELINE` statement can be used with a scatter plot or a histogram.

If a column is used to generate multiple reference lines, then the column type (numeric or string) must agree with the type of data presented on the axis.

**Examples:**

**Example 1: Specifying a Single Reference Line**

This example shows you how to draw a reference line using the `REFERENCELINE` statement. The follow figure shows the output for this example.
Example Program

Here is the SAS code for this example.

```sas
/* Create the template for the graph */
proc template;
    define statgraph referenceline;
        begingraph;
            entrytitle "Line of Symmetry";
            layout overlay / yaxisopts=(linearopts=(viewmin=0));
                seriesplot x=x y=y;
                referenceline x=3 /
                    lineattrs=(color=blue) curvelabel="X=3";
            endlayout;
        endgraph;
    end;
run;

/* Generate the plot data */
data test;
    do X=0 to 8 by 0.25;
        Y=(x-3)*(x-3) + 5;
        output;
    end;
run;

/* Generate the graph */
proc sgrender data=test template=referenceline;
run;
```
Example 2: Specifying Reference Lines Using Data Columns

This example shows you how to specify the reference-line intercept values for multiple reference lines using columns in the data set for the graph. The following figure shows the output for this example.

Output 6.1  Multiple Reference Lines in a Graph

Example Program

Here is the SAS code for this example.

```sas
/* Define the template for the graph */
proc template;
  define statgraph referenceline;
  begingraph;
    entrytitle "Line of Symmetry";
    layout overlay / yaxisopts=(linearopts=(viewmin=0));
      seriesplot x=x y=y;
      referenceline y=yR / curvelabel=label
        lineattrs=(color=gray pattern=dot);
    endlayout;
  endgraph;
end;
run;

/* Generate the graph data */
data graphdata;
  /* Plot data */
  do X=0 to 8 by 0.25;
    /* Plot data */
    do X=0 to 8 by 0.25;
```
Example 3: Specifying Reference Lines Using the COLN and COLC Functions

This example shows you how to use the COLN() and COLC() functions to specify multiple intercept values directly in the REFERENCELINE statement. This approach is an alternative to including the reference line data in the plot data. The output is shown in "Example 2: Specifying Reference Lines Using Data Columns" on page 740.

Example Program

Here is the SAS code for this example.

```sas
/* Create the template for the graph */
proc template;
define statgraph referencelines;
begingroup;
    entrytitle "Line of Symmetry";
    layout overlay / yaxisopts=(linearopts=(viewmin=0));
        seriesplot x=x y=y;
        /* Use COLN() to specify the intercept values */
        referenceline y=eval(coln(14, 5)) / 
            /* Use COLC() to specify a label for each reference line */
            curvelabel=eval(colc("Y=14", "Y=5"))
            lineattrs=(color=gray pattern=dot);
    endgroup;
endgraph;
endgroup;
end;
run;
/* Generate the plot data */
data test;
do X=0 to 8 by 0.25;
    Y=(x-3)*(x-3) + 5;
    output;
end;
run;
/* Generate the graph */
proc sgrender data=test template=referencelines;
run;
```

Example 3: Specifying Reference Lines Using the COLN and COLC Functions

Example Program
REGRESSIONPLOT Statement

Creates a fitted regression line or curve computed from input data.

Restriction: The REGRESSIONPLOT statement supports only models of one independent and one dependent variable.

Tip: Starting with SAS 9.4M3, subpixel rendering is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see “ODS GRAPHICS Statement” on page 1483.

Syntax

REGRESSIONPLOT X=numeric-column | expression
  Y=numeric-column | expression </<regression-options> <options>>;

Summary of Optional Arguments

Appearance options
  DATATRANSPARENCY=number
  INDEX=positive-integer-column | expression
  LINEATTRS=style-element | style-element (line-options) | (line-options)

Axes options
  PRIMARY=TRUE | FALSE
  XAXIS=X | X2
  YAXIS=Y | Y2

Data tip options
  TIPFORMAT=(role-format-list)
  TIPLABEL=(role-label-list)

Label options
  CURVELABEL="string"
specifies a label for the regression line.

**CURVELABELATTRS=** `style-element | style-element (text-options) | (text-options)`

specifies the color and font attributes of the regression line labels.

**CURVELABELLOCATION=** `INSIDE | OUTSIDE`

specifies the location of the regression line label relative to the plot area.

**CURVELABELPOSITION=** `AUTO | MAX | MIN | START | END`

specifies the position of the regression line label relative to the regression line.

**CURVELABELSPLIT=** `TRUE | FALSE`

specifies whether to split the regression line label at the specified split characters.

**CURVELABELSPLITCHAR=** `"character-list"`

specifies one or more characters on which the regression line label can be split if needed.

**CURVELABELSPLITCHARDROP=** `TRUE | FALSE`

specifies whether the split characters are included in the regression line label text.

**CURVELABELSPLITJUSTIFY=** `AUTO | CENTER | LEFT | RIGHT`

specifies the justification of the strings that are inside the regression line label block.

**LEGENDLABEL=** `"string"`

specifies a label to be used in a discrete legend for this plot.

**Midpoint options**

**GROUP=** `column | discrete-attr-var | expression`

creates a distinct set of regression lines from just the observations that correspond to each unique group value of the specified column.

**INCLUDEMISSINGGROUP=** `TRUE | FALSE`

specifies whether missing values of the group variable are included in the plot.

**Plot reference options**

**NAME=** `"string"`

assigns a name to this plot statement for reference in other template statements.

**Required Arguments**

**X=** `numeric-column | expression`

specifies the column for the X values.

**Y=** `numeric-column | expression`

specifies the column for the Y values.

**Optional Arguments**

**CURVELABEL=** `"string"`

specifies a label for the regression line.

**Defaults**

No regression line label is displayed

This option is not valid when the **GROUP=** option is specified.
Tip  
The font and color attributes for the label are specified by the CURVELABELATTRS= option.

**CURVELABELATTRS=**`style-element | style-element (text-options) | (text-options)`  
specifies the color and font attributes of the regression line labels.

**Default**  
The GraphValueText style element.

**Interactions**  
For this option to take effect, the CURVELABEL= option must also be used.

If the GROUP= option is specified, then this option is ignored.

**See**  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Text Options” on page 1525 for available `text-options`.

**CURVELABELLOCATION=**`INSIDE | OUTSIDE`  
specifies the location of the regression line label relative to the plot area.

**INSIDE**  
locates the labels inside the plot area

**OUTSIDE**  
locates the labels outside the plot area

**Default**  
INSIDE

**Restriction**  
OUTSIDE cannot be used when the REGRESSIONPLOT is used in multi-cell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

**Interactions**  
For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELPOSITION= option to determine where the line labels appear.

**See**  
“Location and Position of Curve Labels” on page 200

**CURVELABELPOSITION=**`AUTO | MAX | MIN | START | END`  
specifies the position of the regression line label relative to the regression line.

**AUTO**  
automatically positions the line label near the line boundary along unused axes whenever possible (typically Y2 and X2) in order to avoid collision with tick values.

**Restriction**  
This option is used only when CURVELABELLOCATION= OUTSIDE.

**MAX**  
forces the line label to appear near maximum line values (typically, upper right).

**MIN**  
forces the line label to appear near minimum line values (typically, lower left).
### START
forces the line label to appear near the beginning of the regression line.

**Restriction** This option is used only when CURVELABELLOCATION=INSIDE.

**Tip** This option is particularly useful when the regression line has a spiral shape.

### END
forces the line label to appear near the end of the regression line.

**Restriction** This option is used only when CURVELABELLOCATION=INSIDE.

**Tip** This option is particularly useful when the regression line has a spiral shape.

**Defaults** AUTO when CURVELABELLOCATION=OUTSIDE.

END when CURVELABELLOCATION=INSIDE.

**Restriction** The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

**Interactions** For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELLOCATION= option to determine where the line label appears.

**Note** When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the label might fall outside of the graph area. In that case, the regression-line label might not be displayed or might be positioned incorrectly.

**Tip** When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.

**See** "Location and Position of Curve Labels" on page 200

---

### CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the regression line label at the specified split characters. When a regression line label is split, the label is split on each occurrence of the specified split characters.

**Default** FALSE. The regression line label is not split.

**Requirement** The CURVELABEL= option must also be specified.

**Interactions** The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.

This option has no effect when CURVELABEL=OUTSIDE.
See “boolean” on page 1479 for other Boolean values that you can use.

CURVELABELSPLITCHAR="character-list"
specifies one or more characters on which the regression line label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the regression line label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the regression line label is split unconditionally at each occurrence of any of the specified split characters. If the regression line label does not contain any of the specified characters, then the label is not split.

“character-list”
one or more characters with no delimiter between each character and enclosed in quotation marks.

Default: A blank space

Requirements: The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no delimiters. For example, to specify the split characters a, b, and c, use the following option:

```
curvelabelsplitchar="abc"
```

The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interactions: This option has no effect if CURVELABELPOSITION=AUTO.

The CURVELABELSPLITCHARDROP= option specifies whether the split characters are included in the regression line label or are dropped.

Notes: When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

Tip: Use the CURVELABELSPLITJUSTIFY= option to specify the justification of the strings in the regression line label block.

CURVELABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the regression line label text.

TRUE
drops the split characters from the regression line label text.

FALSE
includes the split characters in the regression line label text. When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.
The following figure shows an example of a regression line label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="\*"

Note: The horizontal line to the left of the label represents the maximum end of the regression line for reference.

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**  TRUE. The split characters are dropped from the regression line label.

**Requirement**  The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction**  The CURVELABELSPLITCHAR= option specifies the split characters.

**See**  "boolean " on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**

specifies the justification of the strings that are inside the regression line label block.

**AUTO**

justifies the labels based on the CURVELABELPOSITION= option, as shown in the following table.

<table>
<thead>
<tr>
<th>CURVELABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>
CENTER
LEFT
RIGHT
justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the regression line for reference.

<table>
<thead>
<tr>
<th>AUTO</th>
<th>CENTER</th>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Group A</td>
<td>Product Group A</td>
<td>Product Group A</td>
<td>Product Group A</td>
</tr>
</tbody>
</table>

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

Default AUTO

Requirement The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction This option has no effect if CURVELABELPOSITION=AUTO.

DATATRANSPARENCY=number
specifies the degree of the transparency of the regression line and line label.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

GROUP=column | discrete-attr-var | expression
creates a distinct set of regression lines from just the observations that correspond to each unique group value of the specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default Each distinct group value might be represented in the plot by a different combination of line color and line pattern. Line colors vary according to the ContrastColor attribute of the GraphData1–GraphDataN and GraphMissing style elements. Line patterns vary according to the LineStyle attribute of the GraphData1–GraphDataN style elements.

Requirement You must sort the data in ascending order by the specified group column before you plot the data. If the plot data is a CAS in-
memory table, you must download the in-memory table to your SAS client, and then sort it locally before you plot it.

**Interactions**

The group values are mapped in the order of the data, unless the `INDEX=` option is used to alter the default sequence of line colors and line patterns.

The `INCLUDEMISSINGGROUP` option controls whether missing group values are considered a distinct group value.

**Note**

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tip**

The `LINEATTRS=` option can be used to override the representations that are used to identify the groups. For example, `LINEATTRS=(PATTERN=SOLID)` can be used to assign the same pattern to all of the lines, letting the line color distinguish group values. Likewise, `LINEATTRS=(COLOR=BLACK)` can be used to assign the same color to all of the lines, letting the line pattern distinguish group values.

See “DISCRETEATTRVAR Statement” on page 1435

**INCLUDEMISSINGGROUP=TRUE | FALSE**

specifies whether missing values of the group variable are included in the plot.

**Default** TRUE

**Interaction** For this option to take effect, the `GROUP=` option must also be specified.

**Tip**

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the `INDEX=` option is used, the `MISSING=` system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See “boolean ” on page 1479 for other Boolean values that you can use.

**INDEX=**positive-integer-column | expression

specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements** The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.
 Interaction  For this option to take effect, the GROUP= option must also be specified.

 Notes  The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

 If you do not use this option, then the group values are mapped in the order of the data.

 Tip  You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

 LEGENDLABEL="string"  specifies a label to be used in a discrete legend for this plot.

 Default  The string specified on the NAME= option.

 Restriction  This option applies only to an associated DISCRETELEGEND statement.

 Interaction  If the GROUP= option is specified, then this option is ignored.

 LINEATTRS=style-element | style-element (line-options) | (line-options)  specifies the attributes of the regression line.

 Default  The GraphFit style element.

 See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

 “Line Options” on page 1522 for available line-options.

 NAME="string"  assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

 Restrictions  The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

 This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

 Interaction  The string is used as the default legend label if the LEGENDLABEL= option is not used.

 PRIMARY=TRUE | FALSE  specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

 Default  FALSE

 Restriction  This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.
Note: In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See: “When Plots Share Data and a Common Axis” on page 979

“boolean” on page 1479 for other Boolean values that you can use.

**TIPFORMAT=(role-format-list)**

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)

a space-separated list of role-name = format pairs.

Example: TIPFORMAT=(Y=6.2)

Default: The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Requirement: To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Note: The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the data tip information.

**TIPLABEL=(role-label-list)**

specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list

a space-separated list of rolename = "string" pairs.

Example: TIPLABEL=(Y="Curve")

Default: The column label or column name of the column assigned to the role.

Note: The columns assigned to the X, Y, and GROUP (if assigned) roles are automatically included in the data tip information.

**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

Interaction: The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Regression Options

**ALPHA=positive-number**
specifies the confidence level to compute.

<table>
<thead>
<tr>
<th>Default</th>
<th>0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>$0 &lt; \text{positive-number} &lt; 1$</td>
</tr>
<tr>
<td>Tip</td>
<td>ALPHA=0.05 represents a 95% confidence level.</td>
</tr>
</tbody>
</table>

**CLI=“name”**
produces confidence limits for individual predicted values for each observation. The confidence level is set by the ALPHA= option.

| Interaction | name is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBand statement. See the example in “Example: REGRESSIONPLOT Statement” on page 753. |

**CLM=“name”**
produces confidence limits for a mean predicted value for each observation. The confidence level is set by the ALPHA= option.

| Interaction | name is a unique name within the template that is case sensitive and cannot contain spaces. It must be assigned in order for the confidence limits to be computed. To display confidence limits, you must use this name as the required argument of a MODELBand statement. See the example in “Example: REGRESSIONPLOT Statement” on page 753. |

**DEGREE=non-negative-integer**
specifies the degree of the polynomial.

<table>
<thead>
<tr>
<th>Default</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranges</td>
<td>0–174 in SAS 9.4M1 and earlier releases.</td>
</tr>
<tr>
<td></td>
<td>1–10 starting with SAS 9.4M2.</td>
</tr>
</tbody>
</table>

**FREQ=numeric-column**
specifies a column in the input data set that represents the frequency of occurrence of the current observation, essentially treating the data set as if each
observation appeared \( n \) times, where \( n \) is the value of the FREQ column for the observation. Noninteger values of the FREQ column are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ column is greater than or equal to 1.

**MAXPOINTS=** *positive-integer*

specifies the maximum number of predicted points generated for the regression curve as well as any confidence limits.

Default 201

**WEIGHT=** *numeric-column*

specifies a column in the input data set that contains values to be used as *a priori* weights for a regression fit. If an observation’s weight is zero, negative, or missing, then the observation is deleted from the analysis.

Interaction Starting with **SAS 9.4M2**, when the CLI= option is used with this option, the confidence band for individual predicted values is displayed as a high-low chart instead of a band.

**Details**

The REGRESSIONPLOT statement supports only models of one independent and one dependent variable. For more information about the fitting methodology, see the TRANSREG procedure in the SAS/STAT user’s guide.

In addition to the regression line, the REGRESSIONPLOT statement can compute confidence levels for the fitted line. To display the confidence levels:

1. Use the CLI= or CLM= regression option(s) to declare a name for each confidence level.

2. Use MODELBAND statements to refer to the name(s) and draw a confidence band(s) from this information.

**Example: REGRESSIONPLOT Statement**

The following graph was generated by the "Example Program" on page 754:
Example Program

```sas
proc template;
    define statgraph regressionplot;
    begingraph;
        entrytitle "Regression Fit Plot";
        layout overlay;
            scatterplot x=weight y=mpg_highway /
                datatransparency=0.7;
            regressionplot x=weight y=mpg_highway /
                name="fitline"
                alpha=0.05 legendlabel="Regression Fit";
            discretelegend "fitline";
        endlayout;
    endgraph;
end;
run;

proc sgrender data=sashelp.cars template=regressionplot;
run;
```

**SCATTERPLOT Statement**

Creates a scatter plot of input data.
Syntax

**SCATTERPLOT** 
\[
X=column \mid \text{expression} \\
Y=column \mid \text{expression} <\text{options}>;
\]

Summary of Optional Arguments

**Appearance options**

- **CLUSTERWIDTH=number**
  - on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

- **COLORMODEL=style-element | (color-list)**
  - specifies a color ramp that is to be used with the COLORRESPONSE= or MARKERCOLORGRADIENT= option.

- **COLORRESPONSE=numeric-column | range-attr-var | expression**
  - starting with SAS 9.4M2, specifies the column or range attribute map variable to use to determine the marker colors.

- **CONTRIBUTE OFFSETS=ALL | NONE | (axis-offset-list)**
  - specifies whether space requirements for this plot contribute to the calculation of the axis offsets.

- **DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**
  - enhances the visual appearance of the plot markers.

- **DATATRANSPARENCY=number**
  - specifies the degree of the transparency of the markers, data labels, and error bars, when displayed.

- **DISCRETEMARKERSIZE=number**
  - specifies the size of a marker as a fraction of the tick spacing.

- **ERRORBARATTRS=style-element | style-element (line-options) | (line-options)**
  - specifies the attributes of the error bars that are associated with the data points.

- **ERRORBARCAPSCALE=positive-number**
  - specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

- **ERRORBARCAPSHAPE=SERIF | NONE**
  - specifies whether the error bars have a serif cap.

- **FILLEDOUTLINEDMARKERS=TRUE | FALSE**
  - specifies whether markers are drawn with both fill and an outline.

- **INDEX=positive-integer-column | expression**
  - specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

- **LABELSTRIP=TRUE | FALSE**
  - specifies whether leading and trailing blanks are stripped from marker characters or data labels that have a fixed position before they are displayed in the plot.

- **MARKERATTRS=style-element | style-element (marker-options) | (marker-options)**
  - specifies the attributes of the data markers.

- **MARKERCHARACTER=column | expression**
  - specifies a column that defines strings that are to be used instead of marker symbols.
MARKERCHARACTERATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the marker character specified on
the MARKERCHARACTER= option.

MARKERCHARACTERPOSITION= CENTER | TOP | BOTTOM | LEFT | RIGHT | TOPLEFT | TOPRIGHT | BOTTOMLEFT | BOTTOMRIGHT
specifies the justification of the marker characters.

MARKERCOLORGRADIENT=numeric-column | range-attr-var | expression
in SAS 9.4M1 and earlier releases, specifies the column or range attribute
map variable that is used to determine the marker colors.

MARKERFILLATTRS=style-element | (fill-options)
specifies the appearance of the filled markers.

MARKEROUTLINEATTRS=style-element | (line-options)
specifies the appearance of the marker outlines.

MARKERSIZEMAX=dimension
for SAS 9.4M1 and for earlier releases, specifies a drawing size for the
largest marker when the marker size represents response values.

MARKERSIZEMIN=dimension
for SAS 9.4M1 and for earlier releases, specifies a drawing size for the
smallest marker when the marker size represents response values.

MARKERSIZERESPONSE=numeric-column | expression
for SAS 9.4M1 and for earlier releases, specifies a column that is used to
map the drawing size of the markers.

OUTLINEDMARKERCHARACTERS=TRUE | FALSE
specifies whether the characters that are used as marker symbols are
outlined in order to enhance their appearance in the graph.

REVERSECOLORMODEL=TRUE | FALSE
specifies whether to reverse the gradient (color ramp) that is defined by
either the ODS style that is in effect or by the COLORMODEL= option.

SIZEMAX=dimension
starting with SAS 9.4M2, specifies a drawing size for the largest marker
when the marker size represents response values.

SIZEMIN=dimension
starting with SAS 9.4M2, specifies a drawing size for the smallest marker
when the marker size represents response values.

SIZERESPONSE=numeric-column | expression
starting with SAS 9.4M2, specifies a column that is used to map the
drawing size of the markers.

SUBPIXEL=AUTO | OFF
specifies whether subpixel rendering is used for image output when the
scatter plot is rendered.

USEDISCRETESIZE=TRUE | FALSE
specifies that the marker size should be based on fraction of the midpoint
spacing that is set by the DISCRETEMARKERSIZE= option.

XERRORLOWER=numeric-column | expression
specifies values for the lower endpoints on the X error bars. The error
bars are drawn from the markers to the endpoints.

XERRORUPPER=numeric-column | expression
specifies values for the upper endpoints on the X error bars.

YERRORLOWER=numeric-column | expression
specifies values for the lower endpoints on the Y error bars.

YERRORUPPER=numeric-column | expression
specifies values for the upper endpoints on the Y error bars.

Axes options

**CLUSTERAXIS**=AUTO | X | Y
specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

**PRIMARY**=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features.

**XAXIS**=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS**=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

**ROLENAME**=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips.

**TIP**=(role-list) | NONE
specifies the information to display when the cursor is positioned over the scatter points.

**TIPFORMAT**=(role-format-list)
specifies display formats for tip columns.

**TIPLABEL**=(role-label-list)
specifies display labels for tip columns.

Label options

**DATALABEL**=column | expression
specifies a column for marker labels. The label positions are adjusted to prevent them from overlapping.

**DATALABELATTRS**=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the data labels.

**DATALABELPOSITION**=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the markers.

**DATALABELSPLIT**=TRUE | FALSE
specifies whether to split the data labels at the specified split characters.

**DATALABELSPLITCHAR**="character-list"
specifies one or more characters on which the data labels can be split if needed.

**DATALABELSPLITCHARDROP**=TRUE | FALSE
specifies whether the split characters are included in the data labels.

**DATALABELSPLITJUSTIFY**=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the data label blocks.

**LEGENDLABEL**="string"
specifies a label to be used in a discrete legend for this plot.

Midpoint options

**DISCRETEOFFSET**=number
specifies an amount to offset all markers from discrete X values, or discrete Y values, or both.

GROUP=column | discrete-attr-var | expression
creates a separate marker type for each unique group value of the specified column.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how marker groups are positioned for the coordinate pairs.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

JITTER=NONE | AUTO
specifies whether to jitter data markers.

JITTEROPTS=(jitter-options)
specifies options for managing jittering.

ODS options
URL=string-column
specifies an HTML page to display when a point is selected.

Plot reference options
NAME="string"
assigns a name to this plot statement for reference in other template statements.

Statistics options
FREQ=numeric-column | expression
specifies a numeric column that provides frequencies for each observation that is read.

Required Arguments

X=column | expression
specifies the column for the X values.

Y=column | expression
specifies the column for the Y values.

Optional Arguments

CLUSTERAXIS=AUTO | X | Y
specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

AUTO
uses the discrete axis for clustering groups when only one axis is discrete.
Uses the X axis for clustering if both axes are discrete or interval.

X
Y
uses the X or Y axis for clustering groups.

Default AUTO
Interaction
The GROUPDISPLAY= option must be set to CLUSTER for this option to have any effect.

CLUSTERWIDTH=number

on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

Default 0.85

Range 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Interactions For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

When GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect for interval data, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

COLORMODEL=style-element | (color-list)
specifies a color ramp that is to be used with the COLORRESPONSE= or MARKERCOLORGRADIENT= option.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR specifies the color for the smallest data value of the COLORRESPONSE= or MARKERCOLORGRADIENT= column.

NEUTRALCOLOR specifies the color for the midpoint of the range of the COLORRESPONSE= or MARKERCOLORGRADIENT= column.
ENDCOLOR specifies the color for the highest data value of the COLORRESPONSE= or MARKERCOLORGRADIENT= column.

(color-list)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

See “color ” on page 1480

Default The ThreeColorAltRamp style element.

Interaction For this option to take effect, the COLORRESPONSE= or MARKERCOLORGRADIENT= option must also be specified.

Tip The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

COLORRESPONSE=numeric-column | range-attr-var | expression
starting with SAS 9.4M2, specifies the column or range attribute map variable to use to determine the marker colors.

Note: Starting with SAS 9.4M2, the COLORRESPONSE= option replaces the MARKERCOLORGRADIENT= option. The syntax and functionality are the same. The MARKERCOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is preferred.

range-attr-var
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

"RANGEATTRVAR Statement"

Restriction A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

Tip For each range in the attribute map, the RANGEALTCOLOR= or RANGEALTCOLORMODEL= option in the RANGE statement determines the marker colors.

See “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

You can use this option to add a second response variable to an analysis. For example, in an analysis of weight by height, you can specify an age column by using the COLORRESPONSE= or MARKERCOLORGRADIENT= option so that the change in the gradient color of the markers reflects the change in age.
Requirement  For a grouped plot, the COLORRESPONSE values should remain constant for each group value. If the COLORRESPONSE column has multiple values for a single GROUP value, unexpected results might occur.

Interactions  When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Suboption COLOR= in the DATALABELATTRS= option overrides this option for the data label color attribute.

This option overrides suboption COLOR= in the MARKERATTRS= option and in the MARKERCHARACTERATTRS= option and varies the marker color according to the color gradient or the attribute map.

Note  You can use MARKERCOLORGRADIENT= as an alternative to COLORRESPONSE=. However, if you use MARKERCOLORGRADIENT=, be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options recognize MARKERCOLORGRADIENT as the color role, not COLORRESPONSE.

Tips  To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

Starting with SAS 9.4M2, when this option is in effect and error bars are displayed, the error bars derive their color from the markers. To set a fixed color for the error bars, use the ERRORBARATTRS= option.

If the MARKERCHARACTER= option is also specified, then the gradients that would be applied to the markers are applied to the text strings.

CONTRIBUTE OFFSETS=ALL | NONE | (axis-offset-list) specifies whether space requirements for this plot contribute to the calculation of the axis offsets.

Note: This feature applies to SAS 9.4M2 and to later releases.
This plot’s layout container queries each of its plots for a preferred offset and includes all of the offsets in the axis offset calculations. If the DATALABEL= or MARKERCHARACTER= option is specified for this plot, this plot might request a preferred offset that prevents the clipping of any data labels or marker character strings that appear at the ends of the axes. The requested offset is based on the longest string. If the label or marker character lengths vary significantly, the result is wasted space when the shorter strings appear near the ends of the axes. In that case, you can use the CONTRIBUTEOFFSETS= option to modify or eliminate this plot’s contribution to the offset calculations in order to reclaim that space.

**ALL**
- the space requirements for this plot are contributed to the axis offset calculations.

**NONE**
- the space requirements for this plot are not contributed to the axis offset calculations.

**(axis-offset-list)**
- a space-delimited list of specific contributions that this plot makes to the axis offset calculations.
  
The list is one or more of the following values, enclosed in parentheses:

**XMAX**
- the space requirements for this plot are contributed to the X-axis offset calculation for the maximum end.

**XMIN**
- the space requirements for this plot are contributed to the X-axis offset calculation for the minimum end.

**YMAX**
- the space requirements for this plot are contributed to the Y-axis offset calculation for the maximum end.

**YMIN**
- the space requirements for this plot are contributed to the Y-axis offset calculation for the minimum end.

**Default**
- **ALL**

**Interaction**
- Offsets that are set in the layout axis options are always honored, regardless of the setting on this option.

**Note**
- This option does not affect offset requests from other plots.

**DATALABEL=column | expression**
- specifies a column for marker labels. The label positions are adjusted to prevent them from overlapping.
Default: No data labels are displayed.

Interactions: If a numeric column is specified and the column has no format, then a BEST6 format is applied. This option is ignored if the MARKERCHARACTER= option is used, which displays labels instead of the markers.

Tip: Use this option to display labels for the markers. The position of the labels is adjusted to prevent the labels from overlapping. If you want labels displayed instead of markers, then use the MARKERCHARACTER= option.

**DATALABELATTRS=** `<style-element> | style-element (text-options) | (text-options)`
specifies the color and font attributes of the data labels.

Defaults: For non-grouped data, the GraphDataText style element. For grouped data, the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

Interactions: For this option to take effect, the DATALABEL= option must also be specified. This option is ignored if the MARKERCHARACTER= option is specified.

Note: When the DATALABELPOSITION=AUTO option is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

See: “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`

"Text Options” on page 1525 for available `text-options`.

**DATALABELPOSITION=** `<AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT>`
specifies the location of the data labels relative to the markers.

**AUTO**
uses a data-label collision avoidance algorithm to position the labels.
See “LABELPLACEMENT=AUTO | GREEDY | SA” on page 33

Note When AUTO is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

TOPRIGHT
TOP
TOPLEFT
LEFT
CENTER
RIGHT
BOTTOMLEFT
BOTTOM
BOTTOMRIGHT
specifies a location for all of the data labels.

Default AUTO

DATALABELSPLIT=TRUE | FALSE
specifies whether to split the data labels at the specified split characters.

Default FALSE

Interactions The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

This option has no effect when DATALABELPOSITION=AUTO.

See “boolean ” on page 1479 for other Boolean values that you can use.

DATALABELSPLITCHAR="character-list"
specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

"character-list"
one or more characters with no space between each character and enclosed in quotation marks.

Default A blank space

Requirements The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

datalabelsplitchar="abc"

Interactions This option has no effect if DATALABELPOSITION=AUTO or if DATALABELSPLIT=FALSE.
The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels. The split characters are specified by the DATALABELSPLITCHAR= option.

**TRUE**

drops the split characters from the data label.

**FALSE**

includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:

- the data label text for this label is Product*Group*A
- DATALABELSPLIT=TRUE
- DATALABELSPLITCHARDROP=TRUE | FALSE
- DATALABELSPLITCHAR="*"

When DATALABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When DATALABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**

TRUE. The split characters are dropped from the data label.

**Interactions**

This option has no effect unless DATALABELSPLIT=TRUE.

The DATALABELSPLITCHAR= option specifies the split characters.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**

specifies the justification of the strings that are inside the data label blocks.
AUTO
justifies the labels based on the DATALABELPOSITION= option.

<table>
<thead>
<tr>
<th>DATALABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

CENTER
LEFT
RIGHT
justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.

Note: The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.

In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

Default AUTO

Interaction This option has no effect if DATALABELPOSITION=AUTO or if DATALABELSPLIT=FALSE.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN enhances the visual appearance of the plot markers. The following figure shows large CIRCLEFILLED markers with each of the skins applied.
The `DATASKIN=` option value that is specified in the `BEGINGRAPH` statement. If not specified, then the `GraphSkins:DataSkin` style element value is used.

Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an `OVERLAY` or `PROTOTYPE` layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the `DATASKINMAX=` option in your ODS GRAPHICS statement to increase the maximum limit.

This option overrides the `BEGINGRAPH` statement `DATASKIN=` option.

When a data skin is applied, all marker outlines are set by the skin, and the outline settings are ignored from the ODS style or from `MARKERATTRS=` option.

### DATATRANSPARENCY=number
specifies the degree of the transparency of the markers, data labels, and error bars, when displayed.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

### DISCRETEMARKERSIZE=number
specifies the size of a marker as a fraction of the tick spacing.

Default 0.5

Range 0 to 1

Requirement For this option to take effect, at least one of the axes must be discrete.

Interactions If both of the axes are discrete, then the marker size is a fraction of the smaller tick spacing.

If the X axis is a numeric interval axis and the `GROUPDISPLAY=CLUSTER` option is in effect, then the marker
size is a fraction of the interval between the two closest but not identical points in the X direction.

For this option to take effect, the `USEDISCRETESIZE=` option must be set to TRUE (the default is FALSE).

**DISCRETEOFFSET=number**

specifies an amount to offset all markers from discrete X values, or discrete Y values, or both. This feature is useful for graphing multiple response variables side by side on a common axis. By default within an overlay-type layout, if a SCATTERPLOT is used with other plots with a discrete axis, then the markers are centered on the discrete X values, or discrete Y values, or both. Depending on the data, the markers might be superimposed over other graph data. The following code fragment shows the default positioning when a SCATTERPLOT is used with a BOXPLOT:

```plaintext
layout overlay / cycleattrs=true
   xaxisopts=(type=discrete);

   scatterplot x=age y=weight;
   boxplot x=age y=weight;

endlayout;
```

To avoid superimposed plots, you can assign a different offset to each plot statement:

```plaintext
layout overlay / cycleattrs=true
   xaxisopts=(type=discrete);

   scatterplot x=age y=weight /
      discreteoffset=0.2;
   boxplot x=age y=weight /
      discreteoffset=-0.2;

endlayout;
```
Default 0 (no offset, all markers are centered on the discrete X values, or discrete Y values, or both)

Range –0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is to the right on discrete X values and up on discrete Y values. If the layout’s axis options set REVERSE=TRUE, then the offset direction is also reversed.

Restriction This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

Tip Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**ERRORBARATTRS=** specifies the attributes of the error bars that are associated with the data points.

**Defaults**

For ungrouped data, the GraphError style element attributes contrastColor, lineStyle, and lineThickness.

For grouped data, the lineStyle and lineThickness attributes from the GraphError style element, and the contrastColor attribute from the GraphData1-GraphDataN style elements.

**Interaction**

For this option to take effect, error bars must be displayed by the XERRORLOWER=, XERRORUPPER=, YERRORLOWER=, or YERRORUPPER= options.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Line Options” on page 1522 for available *line-options*.

**ERRORBARCAPSCALE=** specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

**Note:** This feature applies to SAS 9.4M5 and to later releases.
Default: 1

Interaction: This option is ignored when **ERRORBARCAPSHAPE=NONE** is in effect.

Tips: Use the **ERRORBARCAPSHAPE=** option to specify the cap shape.

The thickness (height) of the SERIF, LINE, and BRACKET is determined by the thickness specified in the **ERRORBARATTRS=** option. The cap line pattern is always solid.

Examples:
- Double the size of the error-bar caps:
  - `errorbarcapscale=2`
- Reduce the size of the error-bar caps by 50%:
  - `errorbarcapscale=0.5`

**ERRORBARCAPSHAPE=SERIF | NONE**

specifies whether the error bars have a serif cap.

<table>
<thead>
<tr>
<th>SERIF</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Defaults: SERIF in **SAS 9.4M5** and earlier releases.

Starting with **SAS 9.4M2**, GraphError:CapStyle style reference. If attribute CapStyle is not defined in the active style, then SERIF is the default value.

Interaction: Starting with **SAS 9.4M5**, when **ERRORBARCAPSCALE=** is in effect, the display width for SERIF caps is the cap’s default width multiplied by the value specified in **ERRORBARCAPSCALE=**.

Tip: The appearance of the error bars is controlled by the **ERRORBARATTRS=** option.

**FILLEDOUTLINEDMARKERS=TRUE | FALSE**

specifies whether markers are drawn with both fill and an outline.

**TRUE**

draws filled markers (marker symbols with the suffix FILLED) using both fill and an outline.

When this option is TRUE, the fill color and outline color for filled markers are determined in the following ways:

- If the GROUP= option is specified, then by default, the fill color is derived from the GraphData1–GraphDataN style elements Color attribute, and the marker outlined color is derived from the GraphData1–GraphDataN style elements ContrastColor attribute.
- If the GROUP= option is not specified, then by default, the fill color is derived from the GraphDataDefault style elements Color attribute, and the marker outlined color is derived from the GraphOutline style elements ContrastColor attribute.
If the COLORRESPONSE= MARKERCOLORGRADIENT= option is specified, then the marker fill is drawn by using the mapped color that is computed from the value of the COLORRESPONSE= or MARKERCOLORGRADIENT= option for that observation. The marker outline is drawn by using the MARKEROUTLINEATTRS= specification.

**FALSE**
draws the markers using fill or an outline, but not both.

**Default** FALSE

**Tip** To specify the marker fill and outline colors for a non-grouped plot, set this option to TRUE, and then use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify the colors.

**See**
- GROUP= on page 771
- MARKERFILLATTRS= on page 781
- MARKEROUTLINEATTRS= on page 782
- COLORRESPONSE= on page 760
- "boolean " on page 1479 for other Boolean values that you can use.

**FREQ=numeric-column | expression**
specifies a numeric column that provides frequencies for each observation that is read.

**Default** All observations have a frequency count of 1.

**Restriction** If the value of the numeric-column is missing or is less than 1, then the observation is not used in the analysis. If the value is not an integer, then only the integer portion is used.

**Note** If \( n \) is the value of the numeric column for a given observation, then that observation is used \( n \) times for the purposes of any statistical computation.

**GROUP=column | discrete-attr-var | expression**
creates a separate marker type for each unique group value of the specified column.

**discrete-attr-var**
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction** A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

**Default** Each distinct group value might be represented in the plot by a different combination of color and marker symbol. Markers vary according to the ContrastColor and MarkerSymbol attributes of the GraphData1–GraphDataN and GraphMissing style elements.
Interactions

The group values are mapped in the order of the data, unless the `INDEX=` option is used to alter the default sequence of markers and colors.

The INCLUDEMISSINGGROUP option controls whether missing group values are considered a distinct group value.

The marker size is set by the `MARKERATTRS=` option.

The `MARKERCHARACTER=`, `COLORRESPONSE=`, and `MARKERCOLORGRADIENT=` options override the group settings for the marker symbol and marker color.

The `SIZERESPONSE=` and `MARKERSIZERESPONSE=` options override this option's `SIZE=` setting.

Note

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

Tip

The representations that are used to identify the groups can be overridden. For example, each distinct group value is represented by a different marker symbol, but the `MARKERATTRS=(SYMBOL=marker)` option could be used to assign the same symbol to all of the plot's marker symbols, letting marker color indicate group values. Likewise, `MARKERATTRS=(COLOR=color)` could be used to assign the same color to all markers, letting marker symbol indicate group values.

See

"DISCRETEATTRVAR Statement" on page 1435

**GROUPDISPLAY=OVERLAY | CLUSTER**

specifies how marker groups are positioned for the coordinate pairs.

**OVERLAY**

draws markers for a given group value at the exact coordinate.

Depending on the data, markers at a given coordinate might overlap.

**CLUSTER**

draws markers for a given group value adjacent to each other.

Default

OVERLAY

Note

When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

Tip

Use the `CLUSTERWIDTH=` option to control the width of the clusters when CLUSTER is in effect.

**GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING**

specifies the ordering of the groups within a category.
orders the groups within a category in the group-column data order.

orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip This option is useful when you want to reverse the category axis.

orders the groups within a category in ascending order.

orders the groups within a category in descending order.

DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

Use the CLUSTERWIDTH= option to control the distance between the group markers in a cluster.

Use the INDEX= option to alter the default sequence of visual attributes that is assigned to the groups.

specifies whether missing values of the group variable are included in the plot.
**INDEX=**positive-integer-column | expression

Specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

**Requirements**
- The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.
- The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.
- All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interactions**
- For this option to take effect, the GROUP= option must also be specified.
- The MARKERCHARACTER=, COLORRESPONSE=, and MARKERCOLORGRADIENT= options override the group settings for the marker symbol and marker color.

**Notes**
- The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.
- If you do not use this option, then the group values are mapped in the order of the data.

**Tip**
- You can use indexing to collapse the number of groups that are represented in a graph. For more information, see "Remapping Groups for Grouped Data" on page 198.

**JITTER=**NONE | AUTO

Specifies whether to jitter data markers.

**NONE**
- Does not offset the data markers.

**AUTO**
- The system determines the best way to display the data markers, based on the settings specified by the JITTEROPTS= option.
The following figures show a side-by-side comparison of a BOX PLOT and SCATTER PLOT in which the JITTER=NONE and JITTER=AUTO options are specified in the SCATTER PLOT statements. The first figure shows the case in which the Y axis is linear and the X axis is discrete.

**One-Dimensional Jittering**

In this case, when jittering is disabled (JITTER=NONE), markers that represent the same Y value are overlaid, which results in some markers being obscured. When jittering is enabled (JITTER=AUTO), one-dimensional systematic jittering occurs along the X axis. The markers that represent approximately the same Y value are offset along the X axis from the midpoint of that value in order to make all of the markers visible.

The next figure shows the case in which both axes are linear.

**Two-Dimensional Jittering**

In this case, when jittering is disabled (JITTER=NONE), markers that represent the same X and Y bin value are overlaid as in the previous case, which results in some markers being obscured. However, unlike the previous case, when jittering is enabled (JITTER=AUTO), two-dimensional random jittering occurs along the X and the Y axes. The markers are offset randomly along both the X and Y axes in order to make all of the markers visible.

Default: NONE
Restrictions

In SAS 9.4, jittering is not supported when GROUPDISPLAY=CLUSTER. Starting with SAS 9.4M1, this restriction is removed.

One-dimensional systematic jittering is not supported when the MARKERCHARACTER= or the MARKERSIZERESPONSE= option is in effect. Random jittering is supported in those cases when both the X and Y axes are interval axes.

Interactions

When jittering is enabled, the FREQ= option is ignored.

If this option is set to AUTO and both axes are discrete, then one-dimensional systematic jittering occurs along the X axis.

Notes

Jittering changes the default axis offsets, but it does not change the axis data range.

When jittering is requested on a discrete axis and a large amount of data is plotted, the jittering process can become resource-intensive. In that case, it might take longer to render the plot.

You should not overlay a jittered scatter plot and a box plot with outliers because their jittering offsets differ. A scatterplot's points are jittered when the numeric values are close enough whereas a box plot's points are jittered only when the numeric values are identical.

JITTEROPTS=(jitter-options)

specifies options for managing jittering. The jitter options can be one or more of the following values, separated by a space:

AXIS=AUTO | X | Y | BOTH

specifies the axis to use for jittering the data markers.

AUTO

the system determines the axis.

The axis is determined as follows:

- If the X axis is discrete, then one-dimensional systematic jittering is applied along the X axis.
- If the Y axis is discrete and the X axis is interval, then one-dimensional systematic jittering is applied along the Y axis.
- If the X and Y axes are interval, then random jittering is applied along both the X and Y axes.

X

jittering is on the X axis.

If the X axis is discrete, then 1-dimensional systematic jittering is applied. Otherwise, 1-dimensional random jittering is applied.

Note If both the X and Y axes are discrete, then specifying BOTH is equivalent to specifying X.

Y

jittering is applied on the Y axis.

If the Y axis is discrete, then 1-dimensional systematic jittering is applied. Otherwise, 1-dimensional random jittering is applied.
**BOTH**
specifies that random jittering is applied on both the X and Y axes. This option applies only when both the X and Y axes are interval.

**Tip** When the X or Y axis is discrete, specifying BOTH is equivalent to specifying AUTO.

**Default** AUTO

**Restriction** When the `MARKERCHARACTER=` or the `MARKERSIZERESPONSE=` option is in effect, 1-dimensional systematic jittering is not supported. Random jittering is supported in those cases when both the X and Y axes are interval axes.

**UNIFORM=TRUE | FALSE**
specifies whether the jitter offsets are uniform across all category values or are varied as needed.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

When the markers for a category value do not fit the jittering space, the jittering offsets are reduced in order to fit the markers for that category value within the jittering space.

**TRUE**
uses a uniform jittering offset across all category values. The offset used is the minimum offset that is required across all of the category values.

**FALSE**
varys the jittering offset only when the markers for a category value do not fit the jittering space. The offset is varied as needed to fit the markers for each case individually.

**Default** FALSE

**Restriction** This option applies to discrete-type jittering only.

**Note** When UNIFORM=TRUE is specified for a scatter plot within a LAYOUT DATALATTICE or LAYOUT DATAPANEL, the jittering is uniform across all potential category and group combinations.

**WIDTH=positive-number**
specifies the width of the jittering space as a fraction of either the midpoint spacing or of the minimal interval width.
Defaults 0.85 for 1-dimensional systematic jittering
0.4 for random jittering on one or both axes.

Note For a discrete axis, this option takes effect only if the jittered markers still overlap. Setting WIDTH= to a higher value increases the jittering space up to the specified width or to a width sufficient to eliminate overlap if it is less than the specified width. In the latter case, increasing the value further has no effect.

Tip The specified number can be greater than 1.

Interaction This option is ignored when JITTER=NONE.

LABELSTRIP=TRUE | FALSE
specifies whether leading and trailing blanks are stripped from marker characters or data labels that have a fixed position before they are displayed in the plot. The MARKERCHARACTER= option specifies the column that provides the marker strings that are to be used in place of marker symbols.

Default FALSE

Interactions This option effects marker strings only when the MARKERCHARACTER= option is specified.

This option effects data labels only when DATALABEL= is specified and DATALABELPOSITION= is not AUTO.

Tip Stripping the blanks from the numeric value strings helps center each string relative to its data point. Stripping is useful when you want to overlay the data values near or inside the markers for a plot.

See “boolean ” on page 1479 for other Boolean values that you can use.

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.

Default The Y-column label. If a label is not defined, then the Y-column name is used.

Restriction This option applies only to an associated DISCRETELEGEND statement.

Interaction If the GROUP= option is specified, then this option is ignored.

MARKERATTRS=style-element | style-element (marker-options) | (marker-options)
specifies the attributes of the data markers.

Defaults For non-grouped data, GraphDataDefault style element.

For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interactions If the COLORRESPONSE= or MARKERCOLORGRADIENT= option is specified, then this option’s COLOR= suboption is ignored.
If the MARKERCHARACTER= option is specified, then this option's SYMBOL= and WEIGHT= suboptions are ignored.

If the MARKERSIZERESPONSE= option is specified, then this option's WEIGHT= suboption is ignored.

If FILLEDOUTLINEDMARKERS=TRUE, then this option's COLOR= suboption is ignored. In that case, to specify the marker fill color, use the MARKERFILLATTRS= option instead.

This option's COLOR= suboption overrides the default behavior for grouped data. When the COLOR= suboption is specified in that case, all markers have the same color, and the marker symbol alone distinguishes the markers.

This option's SYMBOL= suboption overrides the default behavior for grouped data. When the SYMBOL= suboption is specified in that case, all markers have the same symbol, and the symbol color alone distinguishes the markers.

The TRANSPARENCY= fill option overrides this option's DATATRANSPARENCY= suboption.

**Note**

When style-element is specified, only the style element's MARKERSYMBOL, CONTRASTCOLOR, and MARKERSIZE attributes are used.

**See**

"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Marker Options" on page 1523 for available marker-options.

**MARKERCHARACTER=column | expression**

specifies a column that defines strings that are to be used instead of marker symbols.

![Graph](image)

Each string is centered horizontally and vertically at the data point. The data point positions are not adjusted to prevent text overlap.

**Interactions**

This option overrides the DATALABEL= option.

If the GROUP= option is also used, then color is displayed for a DISCRETE legend, but the character is not displayed in the legend.
For text strings, the same colors are applied to the text strings as would have been applied to markers.

One-dimensional systematic jittering is not supported when this option is in effect. Random jittering is supported when this option is in effect and both the X and Y axes are interval axes. See JITTER= on page 774.

Note
If a numeric column is used, then its values are converted to strings using the format associated with the column or using BEST6 if no format is defined.

Tips
Lengthy strings might be clipped by the plot border. To reduce clipping, you can use the OFFSETMIN= and OFFSETMAX= suboptions of the XAXISOPTS= and YAXISOPTS= options to increase the axis offsets.

You can use the MARKERCHARACTERPOSITION= option to change the justification of the marker character.

You can use the OUTLINEDMARKERCHARACTERS= option to enhance the appearance of the marker characters.

You can use the LABELSTRIP= option to strip the leading and trailing blanks from numeric value strings in order to center each string on its data point.

**MARKERCHARACTERATTRS=**

Specifies the color and font attributes of the marker character specified on the MARKERCHARACTER= option.

**Defaults**

For non-grouped data, the GraphDataText style element.

For grouped data, GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

**Interactions**

For this option to take effect, the MARKERCHARACTER= option must also be used.

When the GROUP= option is also specified, each distinct group value might be represented by a different color (depending on the ODS style setting or the setting on the INDEX= option). The marker character that is associated with the group is assigned the group color.

This option’s COLOR= suboption can be used to specify a single color for all marker characters in a grouped plot, without affecting items that have a group color, such as error bars and marker symbols.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.
MARKERCHARACTERPOSITION=CENTER | TOP | BOTTOM | LEFT | RIGHT | TOPLEFT | TOPRIGHT | BOTTOMLEFT | BOTTOMRIGHT

specifies the justification of the marker characters. The following figure shows the effect of each of the values on the position of marker character M.

```
<table>
<thead>
<tr>
<th>CENTER</th>
<th>TOP</th>
<th>BOTTOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>LEFT</td>
<td>RIGHT</td>
<td>TOLEFT</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>TOPRIGHT</td>
<td>BOTTOMLEFT</td>
<td>BOTTOMRIGHT</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>
```

Note: The red dots represent the center in each case for reference.

Default: CENTER

Interaction: This option is ignored if the MARKERCHARACTER= option is not specified.

Tip: You can use the LABELSTRIP= option to strip the leading and trailing blanks from value strings in order to properly justify each string on its data point.

MARKERCOLORGRADIENT=numic-column | range-attr-var | expression

in SAS 9.4M1 and earlier releases, specifies the column or range attribute map variable that is used to determine the marker colors.

Note: Starting with SAS 9.4M2, the MARKERCOLORGRADIENT= option is deprecated and replaced with the COLORRESPONSE= option. The syntax and functionality are the same. The MARKERCOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is preferred.

Note: Starting with SAS 9.4M2, if you use the MARKERCOLORGRADIENT= option, then be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options recognize the MARKERCOLORGRADIENT role and not the COLORRESPONSE role.

See: COLORRESPONSE= on page 760

MARKERFILLATTRS=style-element | (fill-options)

specifies the appearance of the filled markers.

Defaults: For non-grouped data, the COLOR attribute of the GraphDataDefault style element

For grouped data, the COLOR attribute of a GraphData1–GraphDataN style element
Restriction  The TRANSPARENCY= fill option is ignored. Use the MARKERATTRS= option to set the marker transparency.

Interactions  This option is in effect only when FILLEDOUTLINEDMARKERS=TRUE and the DISPLAY= option enables fill display.

When the COLORRESPONSE= or MARKERCOLORGRADIENT= option is in effect, this option's COLOR= specification is ignored.

Note  When style-element is specified, only the style element's COLOR attribute is used.

See  “General Syntax for Attribute Options” on page 1519

“Fill Color Options” on page 1520

MARKEROUTLINEATTRS=style-element | (line-options)

specifies the appearance of the marker outlines.

Defaults  For non-grouped data, the GraphOutlines style element.

For grouped data, the LineThickness attribute of the GraphOutlines style element and the ContrastColor attribute of a GraphData1–GraphDataN style element.

Restriction  The line style of the marker outline is always solid.

Interaction  This option is ignored when a data skin is applied by the current style or by the DATASKIN= option. In the latter case, the outline is set by the data skin.

Note  When style-element is specified, only the style element's CONTRASTCOLOR and LINETHICKNESS attributes are used.

See  “General Syntax for Attribute Options” on page 1519

“Line Options” on page 1522

MARKERSIZEMAX=dimension

for SAS 9.4M1 and for earlier releases, specifies a drawing size for the largest marker when the marker size represents response values.

Note: Starting with SAS 9.4M2, the SIZEMAX= option replaces the MARKERSIZEMAX= option. The syntax and functionality are the same. The MARKERSIZEMAX= option is still honored, but the SIZEMAX= option is preferred.

See  SIZEMAX= on page 785

“dimension” on page 1480

MARKERSIZEMIN=dimension

for SAS 9.4M1 and for earlier releases, specifies a drawing size for the smallest marker when the marker size represents response values.
Note: Starting with SAS 9.4M2, the SIZEMIN= option replaces the MARKERSIZEMIN= option. The syntax and functionality are the same. The MARKERSIZEMIN= option is still honored, but the SIZEMIN= option is preferred.

See SIZEMIN= on page 785

"dimension" on page 1480

MARKERSIZERESPONSE=numeric-column | expression
for SAS 9.4M1 and for earlier releases, specifies a column that is used to map the drawing size of the markers.

Note: Starting with SAS 9.4M2, the SIZERESPONSE= option replaces the MARKERSIZERESPONSE= option. The syntax and functionality are the same. The MARKERSIZERESPONSE= option is still honored, but the SIZERESPONSE= option is preferred.

Interaction One-dimensional systematic jittering is not supported when this option is in effect. Random jittering is supported when this option is in effect and both the X and Y axes are interval axes. See JITTER= on page 774.

NAME="string"
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction The string is used as the default legend label if the LEGENDLABEL= option is not used.

OUTLINEDMARKERCHARACTERS=TRUE | FALSE
specifies whether the characters that are used as marker symbols are outlined in order to enhance their appearance in the graph.

Note: This option is valid starting in SAS 9.4M1.

Note: This feature is deprecated starting with SAS 9.4M2. The OUTLINEDMARKERCHARACTERS= option is still honored, but the TEXTPLOT statement is now the preferred method of creating a scatter plot using text markers.

The following figure shows the marker characters M and F displayed when OUTLINEDMARKERCHARACTERS=FALSE (default) and when OUTLINEDMARKERCHARACTERS=TRUE.
### OUTLINEDMARKERCHARACTERS

<table>
<thead>
<tr>
<th>FALSE</th>
<th>TRUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>![M F]</td>
<td>![M F]</td>
</tr>
</tbody>
</table>

**Default**  
FALSE

**Restriction**  
Outline marker characters are not supported in vector graphics output. When this option is set and vector graphics output is requested, the graph is converted into an image instead. A note indicating the conversion is written to the SAS log. To restore the vector graphics output in that case, remove the OUTLINEDMARKERCHARACTERS= option from the SCATTERPLOT statement.

**Interaction**  
The MARKERCHARACTER= option must be specified for this option to have any effect.

**See**  
“boolean ” on page 1479 for other Boolean values that you can use.

### PRIMARY=TRUE | FALSE

specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**  
FALSE

**Restriction**  
This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note**  
In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

**See**  
“When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

### REVERSECOLORMODEL=TRUE | FALSE

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

**Default**  
FALSE

**See**  
COLORMODEL=  
“boolean ” on page 1479 for other Boolean values that you can use.

### ROLENAME=(role-name-list)

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.
(role-name-list)
a space-separated list of role-name = column pairs.

Example
The following example assigns the column Obs to the user-defined role TIP:
ROLENAME=(TIP=OBS)

Default
No user-defined roles

Requirement
The role names that you choose must be unique and different from the predefined roles X, Y, DATALABEL, MARKERCHARACTER, COLORRESPONSE or MARKERCOLORGRADIENT, XERRORLOWER, XERRORUPPER, YERRORLOWER, YERRORUPPER, GROUP, and INDEX.

SIZEMAX=dimension
starting with SAS 9.4M2, specifies a drawing size for the largest marker when the marker size represents response values.

Note: Starting with SAS 9.4M2, the SIZEMAX= option replaces the MARKERSIZEMAX= option. The syntax and functionality are the same. The MARKERSIZEMAX= option is still honored, but the SIZEMAX= option is preferred.

Default
Three times the GraphDataDefault:MarkerSize setting (typically 7px) or 21px.

Restriction
The SIZEMAX= value must be greater than the SIZEMIN= or MARKERSIZEMIN= value.

Interactions
For this option to take effect, the SIZERESPONSE= or MARKERSIZERESPONSE= option must also be used.

Tip
If you specify this size as a percent, then the specified value is interpreted as a percent of the graph’s height. You can control the height by using the DESIGNHEIGHT= option of the BEGINGRAPH statement, or by using the HEIGHT= option of the ODS GRAPHICS statement. For a standard 640px by 480px output size, a percentage value of 4.5% sets a maximum size of about 21px, which is approximately the same marker size that would result from this option’s typical default setting.

See
"dimension" on page 1480

SIZEMIN=dimension
starting with SAS 9.4M2, specifies a drawing size for the smallest marker when the marker size represents response values.

Note: Starting with SAS 9.4M2, the SIZEMIN= option replaces the MARKERSIZEMIN= option. The syntax and functionality are the same. The MARKERSIZEMIN= option is still honored, but the SIZEMIN= option is preferred.
The GraphDataDefault:MarkerSize setting, which is typically 7px.

Restriction

The SIZEMIN= value must be less than the SIZEMAX= or MARKERSIZEMAX= value.

Interactions

For this option to take effect, you must also specify the SIZERESPONSE= or MARKERSIZERESPONSE= option.

This option is ignored when the MARKERCHARACTER= option is used. To manage the size of marker characters in a scatter plot, use the TEXTPLOT statement.

Tip

If you specify this size as a percent, then the specified value is interpreted as a percent of the graph’s height. You can control the height by using the DESIGNHEIGHT= option of the BEGINGRAPH statement, or by using the HEIGHT= option of the ODS GRAPHICS statement. For a standard 640px by 480px output size, a percentage value of 1.5% sets a minimum size of about 7px, which is approximately the same marker size that would result from this option’s typical default setting.

See

"dimension" on page 1480

SIZERESPONSE=numeric-column | expression

starting with SAS 9.4M2, specifies a column that is used to map the drawing size of the markers.

Note: Starting with SAS 9.4M2, the SIZERESPONSE= option replaces the MARKERSIZERESPONSE= option. The syntax and functionality are the same. The MARKERSIZERESPONSE= option is still honored, but the SIZERESPONSE= option is preferred.

By default, the minimum and maximum values of this column establish a range over which the marker sizes vary in linear proportion. The actual drawing size of the smallest marker and the largest marker is set automatically.

Default

The GraphDataDefault:MarkerSize setting, which is typically 7px.

Interactions

This option overrides the SIZE= setting in the MARKERATTRS= option.

This option is ignored when the MARKERCHARACTER= option is used. To manage the size of marker characters in a scatter plot, use the TEXTPLOT statement.

Tip

You can adjust the smallest and largest marker size with the SIZEMIN= and SIZEMAX= options, or with the MARKERSIZEMIN= and MARKERSIZEMAX= options.

SUBPIXEL=AUTO | OFF

specifies whether subpixel rendering is used for image output when the scatter plot is rendered.

Note: This feature applies to SAS 9.4M3 and to later releases.
AUTO
sets subpixel rendering OFF, unless SUBPIXEL=ON is specified in the
BEGINGRAPH statement or in an ODS GRAPHICS statement. In that case,
subpixel rendering is ON.

OFF
disables subpixel rendering for image output only for this scatter plot.

Default  AUTO

Requirement Anti-aliasing must be enabled for this option to have any effect.

Notes This option affects subpixel rendering only for this plot. Subpixel
rendering for other plots in the graph is not affected.

For vector-graphics output, this option is ignored, and subpixel
rendering is always enabled.

When subpixel rendering is used for the graph but is turned OFF
for this scatter plot, some elements in the scatter plot such as the
plot markers might be offset a half pixel. This can make the plot
markers appear blurry in the image output.

Tips Anti-aliasing is enabled by default. If anti-aliasing has been
disabled, use the ANTIALIAS=ON option in the ODS GRAPHICS
statement to re-enable it.

Anti-aliasing is disabled automatically for this plot when the
resources required to anti-alias it exceed a preset threshold. When
anti-aliasing is disabled for this or any other plot in the graph,
subpixel rendering is disabled for the entire graph. A note is written
to the SAS log that provides information about how to use the
ANTIALIASMAX= option in an ODS GRAPHICS statement to re-
enable anti-aliasing.

See “Using Subpixel Rendering” in SAS Graph Template Language:
User’s Guide

“ODS GRAPHICS Statement” on page 1483 for information about
the ANTIALIAS=, ANTIALIASMAX=, and SUBPIXEL= options.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over the scatter
points. If this option is used, then it replaces all of the information that is
displayed by default. Roles for columns that do not contribute to the scatter plot
can be specified along with roles that do.

(role-list)
an ordered, space-separated list of unique SCATTERPLOT and user-defined
roles.

SCATTERPLOT roles include X, Y, DATALABEL, MARKERCHARACTER,
COLORRESPONSE or MARKERCOLORGRADIENT, XERRORLOWER,
XERRORUPPER, YERRORLOWER, YERRORUPPER, GROUP, and
INDEX.

User-defined roles are defined with the ROLENAME= option.

Requirement If you use the COLORRESPONSE= option, you must specify
the COLORRESPONSE role for the color values. Likewise, if
you use the MARKERCOLORGRADIENT= option, you must specify the MARKERCOLORGRADIENT role for the color values. Although they are functionally the same, you cannot mix the COLORRESPONSE= and MARKERCOLORGRADIENT= options, and their corresponding roles.

Example

The following example displays data tips for the columns assigned to the roles X, XERRORUPPER, and XERRORLOWER, as well as the column Obs, which is not assigned to any pre-defined SCATTERPLOT role. The Obs column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X XERRORUPPER XERRORLOWER)
```

**NONE**

suppresses data tips and URLs (if requested) from the plot.

**Default**

The columns assigned to these roles are automatically included in the data tip information: X, Y, DATALABEL, MARKERCHARACTER, COLORRESPONSE or MARKERCOLORGRADIENT, XERRORLOWER, XERRORUPPER, YERRORLOWER, YERRORUPPER, FREQ, and GROUP.

**Requirement**

To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interaction**

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tip**

The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

**TIPFORMAT=(role-format-list)**

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)

a space-separated list of role-name = format pairs.

Example

```
ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
```

**Default**

The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**

Only the roles that appear in the TIP= option are used.

**Requirement**

A column must be assigned to each of the specified roles. (See the ROLENAME= option.)
TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename = "string" pairs.

Example
ROLENANE=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

Default
The column label or column name of the column assigned to the role.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

URL=string-column
specifies an HTML page to display when a point is selected.

string-column
specifies a column that contains a valid HTML page reference (HREF) for each marker that is to have an active link.

Example
http://www.sas.com

Requirement
To generate selectable markers, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interactions
This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tips
The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding point is selected.

The URL value can be the same for any X and Y pairs. In that case, the same action is taken when the points for those X and Y pairs are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

USEDISCRETESIZE=TRUE | FALSE
specifies that the marker size should be based on fraction of the midpoint spacing that is set by the DISCRETEMARKERSIZE= option.

Default FALSE
See `DISCRETEMARKERSIZE=`

“boolean " on page 1479 for other Boolean values that you can use.

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

- **Default**  
  X

- **Interactions**  
  This option is ignored if the X= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**XERRORLOWER=numeric-column | expression**
specifies values for the lower endpoints on the X error bars. The error bars are drawn from the markers to the endpoints.

- **Default**  
  The lower segment of the error bars is not drawn.

- **Note**  
  The values are actual values, not relative values.

- **Tips**  
  The appearance of the error bars is controlled by the `ERRORBARATTRS=` option.

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the `MARKERATTRS=` option to specify a degree of transparency for the filled markers.

**XERRORUPPER=numeric-column | expression**
specifies values for the upper endpoints on the X error bars. The error bars are drawn from the markers to the endpoints.

- **Default**  
  The upper segment of the error bars is not drawn.

- **Note**  
  The values are actual values, not relative values.

- **Tips**  
  The appearance of the error bars is controlled by the `ERRORBARATTRS=` option.

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the `MARKERATTRS=` option to specify a degree of transparency for the filled markers.

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

- **Default**  
  Y

- **Interactions**  
  This option is ignored if the Y= argument is not specified.
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YERRORLOWER=numeric-column | expression**
specifies values for the lower endpoints on the Y error bars. The error bars are drawn from the markers to the endpoints.

- **Default**: The lower segment of the error bars is not drawn.
- **Note**: The values are actual values, not relative values.
- **Tips**: The appearance of the error bars is controlled by the `ERRORBARATTRS=` option.

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the `MARKERATTRS=` option to specify a degree of transparency for the filled markers.

**YERRORUPPER=numeric-column | expression**
specifies values for the upper endpoints on the Y error bars. The error bars are drawn from the markers to the endpoints.

- **Default**: The upper segment of the error bars is not drawn.
- **Note**: The values are actual values, not relative values.
- **Tips**: The appearance of the error bars is controlled by the `ERRORBARATTRS=` option.

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the `MARKERATTRS=` option to specify a degree of transparency for the filled markers.

**Examples:**

Example 1: Grouped Scatter Plot

The following graph was generated by the “Example Program” on page 792:
Example Program

```plaintext
proc template;
  define statgraph scatterplot;
    begingraph;
      entrytitle "Height and Weight by Sex";
      layout overlay;
        scatterplot x=height y=weight / group=sex name="scatter" datalabel=name;
        discretelegend "scatter";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=sashelp.class template=scatterplot;
run;
```

Example 2: Discrete Scatter Plot
The following graph was generated by the “Example Program” on page 793:
Example Program

/* Summarize SASHELP.CARS for mean MPG_HIGHWAY */
proc summary data=sashelp.cars nway;
   class type;
   var mpg_highway;
   output out=mileage mean=mean;
run;

/* Sort by mileage */
proc sort data=mileage;
   by mean;
run;

/* Define the graph template */
proc template;
   define statgraph scatterplot;
      begingraph;
         entrytitle "Average Highway MPG By Type";
         layout overlay /
            xaxisopts=(griddisplay=on gridattrs=(color=lightgray))
            yaxisopts=(griddisplay=on gridattrs=(color=lightgray)
               linearopts=(minortickcount=9
                  minorgridattrs=(color=lightgray pattern=dot)));
         scatterplot x=type y=mean /
            group=type groupdisplay=cluster
            markerattrs=graphDataDefault;
      endlayout;
   endgraph;
end;
run;
/* Render the graph */
proc sgrender data=mileage template=scatterplot;
run;

Details
This example creates a scatter plot of average highway mileage by vehicle type. The SUMMARY procedure is used to compute the mean highway mileage for each vehicle type. The SORT procedure is then used to sort the data by mileage in ascending order. Normally, when you plot discrete values using the SCATTERPLOT statement, the values at each end of the category axis are offset to accommodate the width of the largest marker. When a small symbol is used, this can result in the minimum and maximum symbols being placed very close to the edge of the axis. You can use the OFFSETMAX= and OFFSETMIN= axis options to specify offsets for each end of the category axis. However, you have to determine an appropriate offset value.

In this example, the GROUP= and GROUPDISPLAY= options are used to create offsets automatically. The GROUP= option creates a group for each category value. The GROUPDISPLAY=CLUSTER option displays the group values in a cluster and automatically offsets the end values by one-half of the available midpoint spacing, which is similar to the offsets that are used in bar charts. To restore the default colors and symbols that are used for ungrouped data, the MARKERATTRS= option is used to set the marker attributes to the graphDataDefault style element.

Example 3: Dot Plot
The following graph was generated by the "Example Program" on page 793:

![Mean Height by Age](image)
Example Program

```sas
proc summary data=sashelp.class nway;
  var height;
  class age;
  output out=classSum mean=mean std=std;
run;

proc template;
  define statgraph dotplot;
  begingraph;
    entrytitle "Mean Height by Age";
    layout overlay /
      xaxisopts=(label="Height (Mean)")
      yaxisopts=(griddisplay=on
        gridattrs=(color=lightgray pattern=dot)
        reverse=true);
    scatterplot x=mean y=age / name="scatter"
      markerattrs=(symbol=circlefilled)
      xerrorupper=eval(mean + std)
      xerrorlower=eval(mean - std)
      legendlabel="Height (Mean), 1 Standard Deviation";
    discretelegend "scatter";
  endlayout;
  endgraph;
end;
run;

proc sgrender data=classSum template=dotplot;
  where age < 16;
run;
```

Details

This example uses the SCATTERPLOT statement to create a dot plot from data computed by the SUMMARY procedure. The SUMMARY procedure computes the mean and standard deviation for variable Height by variable Age in data set Sashelp.Class, and stores the results in data set Work.classSum. In the graph template code, the SCATTERPLOT statement plots variables Mean and Age in data set Work.classSum. The MARKERATTRS= option specified filled circles as the plot dots. The XERRORUPPER= and XERRORLOWER= options specify the statistic limits. The EVAL function computes the limits from variables Mean and Std.

SCATTERPLOTMATRIX Statement

Creates a matrix of all pairwise scatter plots of the specified columns.

Restriction: The SCATTERPLOTMATRIX statement cannot appear within a LAYOUT OVERLAY, LAYOUT OVERLAY3D, or LAYOUT OVERLAYEQUATED block. It is typically placed in a LAYOUT GRIDDED block.
Syntax

SCATTERPLOTMATRIX numeric-column-list <options>;

Summary of Optional Arguments

Appearance options

COLORMODEL=style-element | (color-list)
  specifies a color ramp that is to be used with the COLORRESPONSE= or MARKERCOLORGRADIENT= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
  starting with SAS 9.4M2, specifies the column or range attribute map variable to use to determine the marker colors.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
  enhances the visual appearance of the plot markers.

DATATRANSPARENCY=number
  specifies the degree of the transparency of the markers and data labels.

DIAGONAL=LABEL | (graph-list)
  specifies whether the diagonal cells of the matrix are labeled with the labels (names) of the required arguments, or with a plot.

INDEX=positive-integer-column | expression
  specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

LABELSTRIP=TRUE | FALSE
  specifies whether leading and trailing blanks are stripped from marker characters or data labels that have a fixed position before they are displayed in the plot.

MARKERATTRS=style-element | style-element (marker-options) | (marker-options)
  specifies the attributes of the data markers.

MARKERCHARACTER=column | expression
  specifies a column that defines strings that are to be used instead of marker symbols.

MARKERCHARACTERATTRS=style-element | style-element (text-options) | (text-options)
  specifies the color and font attributes of the marker character specified on the MARKERCHARACTER= option.

MARKERCHARACTERPOSITION=_CENTER | TOP | BOTTOM | LEFT | RIGHT | TOPLEFT | TOPRIGHT | BOTTOMLEFT | BOTTOMRIGHT
  specifies the justification of the marker characters.

MARKERCOLORGRADIENT=numeric-column | range-attr-var | expression
  in SAS 9.4M1 and earlier releases, specifies the column or range attribute map variable that is used to determine the marker colors.

MATRIXTYPE=FULL | UPPERTRIANGLE | LOWERTRIANGLE
  specifies whether to display the full matrix, or just the upper or lower triangle of the matrix.

REVERSECOLORMODEL=TRUE | FALSE
  specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

START=TOPLEFT | BOTTOMLEFT
  specifies whether to start populating the matrix from the top left or bottom left corner.
SUBPIXEL=AUTO | OFF
specifies whether subpixel rendering is used for image output when the
scatter plots are rendered.

WALLCOLOR=style-reference | color
specifies the fill color of the plot wall area.

WALLDISPLAY=STANDARD | ALL | NONE | (display-options)
specifies whether the plot’s wall and wall outline are displayed.

Confidence options
ELLIPSE=(<ellipse-suboptions>)
specifies that a confidence ellipse be included in each cell containing a
scatter plot.

Data tip options
ROLENAMESPACE=(role-name-list)
specifies user-defined roles that can be used to display information in the
data tips.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over the
scatter points.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns.

TIPLABEL=(role-label-list)
specifies display labels for tip columns.

Inset options
INSET=(info-options)
specifies what information is displayed in an inset.

INSETOPTS=(appearance-options)
specifies location and appearance options for the inset information.

Label options
DATALABEL=column
specifies a column for marker labels.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the data labels.

DATALABELPOSITION=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the markers.

DATALABELSPLIT=TRUE | FALSE
specifies whether to split the data labels at the specified split characters.

DATALABELSPLITCHAR="character-list"
specifies one or more characters on which the data labels can be split if needed.

DATALABELSPLITCHARDROP=TRUE | FALSE
specifies whether the split characters are included in the data labels.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the data label
blocks.

Midpoint options
GROUP=column | discrete-attr-var | expression
creates a distinct set of scatter markers, error bars, and data labels for each unique group value of the specified column.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

**Plot reference options**

**NAME="string"**
assigns a name to this plot statement for reference in other template statements.

**Statistics options**

**CORROPTS=(correlation-options)**
specifies options for computing measures of association between pairs of columns.

**FREQ=numeric-column | expression**
specifies a numeric column that provides frequencies for each observation that is read.

**ROWVARS=(numeric-column-list)**
specifies a secondary list of numeric columns to be paired with the required column list that is specified by the SCATTERPLOTMATRIX statement.

**Required Argument**

**numeric-column-list**
specifies a list of numeric columns to plot.

**Requirements**
There must be at least two columns to produce a useful matrix.

All of the columns must be numeric.

**Note**
The default width is 640px, and the default height is 480px. The graph size is not automatically adjusted to accommodate a large number of columns.

**Tips**
To change the graph size for the current template, use the DESIGNHEIGHT= and DESIGNWIDTH= options in the BEGINGRAPH statement.

To change the graph size for all templates in the current SAS session, use the HEIGHT= and WIDTH= options in the ODS GRAPHICS statement. Size settings in the ODS GRAPHICS statement override size settings in the BEGINGRAPH statement.

You can also limit the number of columns in the matrix (perhaps to seven in each dimension, for example) so that the resulting graphs are not too small to be useful.

**Optional Arguments**

**COLORMODEL=style-element | (color-list)**
specifies a color ramp that is to be used with the COLORRESPONSE= or MARKERCOLORGRADIENT= option.
**style-element**

specifies the name of a style element.

The style element should contain these style attributes:

- **STARTCOLOR** specifies a color for the smallest data value of the COLORRESPONSE= or MARKERCOLORGRADIENT= column.
- **NEUTRALCOLOR** specifies a color for the midpoint of the range of the COLORRESPONSE= or MARKERCOLORGRADIENT= column.
- **ENDCOLOR** specifies a color for the highest data value of the COLORRESPONSE= or MARKERCOLORGRADIENT= column.

**(color-list)**

specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement**
The list of colors must be enclosed in parentheses.

**See**

"color " on page 1480

**Default**
The ThreeColorAltRamp style element.

**Interactions**

For this option to take effect, the COLORRESPONSE= or MARKERCOLORGRADIENT= option must also be specified.

The REVERSECOLORMODEL= option can be used to reverse the start and end colors of the ramp assigned to the color model.

**COLORRESPONSE=**

**numeric-column | range-attr-var | expression**

starting with SAS 9.4M2, specifies the column or range attribute map variable to use to determine the marker colors.

**Note:** Starting with SAS 9.4M2, the COLORRESPONSE= option replaces the MARKERCOLORGRADIENT= option. The syntax and functionality are the same. The MARKERCOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is preferred.

**range-attr-var**

specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

"RANGEATTRVAR Statement"

**Restriction**

A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

**Tip**

For each range in the attribute map, the RANGEALTCOLOR= or RANGEALTCOLORMODEL= option in the RANGE statement determines the marker colors.
When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

You can use this option to add a second response variable to an analysis. For example, in an analysis of weight by height, you can specify an age column by using the COLORRESPONSE= or MARKERCOLORGRADIENT= option so that the change in the gradient color of the markers reflects the change in age.

For a grouped plot, the COLORRESPONSE values should remain constant for each group value. If the COLORRESPONSE column has multiple values for a single GROUP value, unexpected results might occur.

When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Suboption COLOR= in the DATALABELATTRS= option overrides this option for the data label color attribute.

Suboption COLOR= in the MARKERATTRS= option or in the MARKERCHARACTERATTRS= option overrides this option for the marker colors.

You can use MARKERCOLORGRADIENT= as an alternative to COLORRESPONSE=. However, if you use MARKERCOLORGRADIENT=, be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options recognize MARKERCOLORGRADIENT as the color role, not COLORRESPONSE.

To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

If the MARKERCHARACTER= option is also specified, then the gradients that would be applied to the markers are applied to the text strings.
CORROPTS=(*correlation-options*)
  specifies options for computing measures of association between pairs of columns. The following *correlation-options* are available:

EXCLNPWGT=TRUE | FALSE
  specifies whether observations with non-positive weight values are excluded (TRUE) from the analysis.
  Default FALSE (observations with negative weights are treated like those with zero weights and counted in the total number of observations).
  See “boolean ” on page 1479 for other Boolean values that you can use.

NOMISS=TRUE | FALSE
  specifies whether observations with missing values are excluded (TRUE) from the analysis.
  Default FALSE (correlation statistics are computed using all of the nonmissing pairs of columns).
  Note Using NOMISS=TRUE is computationally more efficient.
  See “boolean ” on page 1479 for other Boolean values that you can use.

WEIGHT=numeric-column
  specifies a weighting column to use in the calculation of Pearson weighted product-moment correlation.
  The observations with missing weights are excluded from the analysis. If you use this WEIGHT correlation option, then consider which value of the VARDEF=correlation option is appropriate.
  Default For observations with non-positive weights, the weights are set to zero and the observations are included in the analysis.
  Tip You can include EXCLNPWGT among the correlation options to exclude observations with negative or zero weights from the analysis.

VARDEF=DF | N | WDF | WEIGHT
  specifies the variance divisor in the calculation of variances and covariances.
  DF
    Degrees of Freedom (N – 1)
  N
    number of observations
  WDF
    sum of weights minus 1 (WEIGHT – 1)
  WEIGHT
    sum of weights
  Default DF
  Interaction This option has effect only when the INSET= option is also used.
  See the CORR procedure information in *Base SAS Procedures Guide: Statistical Procedures* for the statistical and computational details of these options.
**DATALABEL=column**
specifies a column for marker labels. The label positions are adjusted to prevent the labels from overlapping.

**Interactions**
If a numeric column is specified and the column has no format, then a BEST6 format is applied.

This option is ignored if the **MARKERCHARACTER=** option is used.

**Note**
The position of the labels are adjusted to prevent the labels from overlapping.

**DATALABELATRBS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the data labels.

**Defaults**
For non-grouped data, the GraphDataText style element.

For grouped data, the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

**Interactions**
For this option to take effect, the **DATALABEL=** option must also be specified.

This option is ignored if the **MARKERCHARACTER=** option is specified.

**Note**
When the DATALABELPOSITION=AUTO option is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

**See**
"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Text Options" on page 1525 for available text-options.

**DATALABELPOSITION=**AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies the location of the data labels relative to the markers.

**AUTO**
uses a data-label collision avoidance algorithm to position the labels.

See "LABELPLACEMENT=AUTO | GREEDY | SA" on page 33

**Note**
When AUTO is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

**TOPRIGHT**
**TOP**
**TOPLEFT**
**LEFT**
**CENTER**
**RIGHT**
**BOTTOMLEFT**
**BOTTOM**
**BOTTOMRIGHT**
specifies a location for all of the data labels.
**DATALABELSPLIT=TRUE | FALSE**

specifies whether to split the data labels at the specified split characters.

Default: FALSE. The data labels are not split.

**Requirement**
The DATALABEL= option must also be specified.

**Interactions**
The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

This option has no effect when DATALABELPOSITION=AUTO.

**See**
“boolean ” on page 1479 for other Boolean values that you can use.

**DATALABELSPLITCHAR="character-list"**
specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

"character-list"  
one or more characters with no space between each character and enclosed in quotation marks.

Default: A blank space

**Requirements**
The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

datalabelsplitchar="abc"

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interactions**
This option has no effect if DATALABELPOSITION=AUTO.

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**
When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**
Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.
**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels.

**TRUE**
- drops the split characters from the data label.

**FALSE**
- includes the split characters in the data label.

When **DATALABELSPLIT=TRUE** and **DATALABELSPLITCHARDROP=FALSE**, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:
- the data label text for this label is `Product*Group*A`
- **DATALABELSPLIT=TRUE**
- **DATALABELSPLITCHARDROP=TRUE | FALSE**
- **DATALABELSPLITCHAR=***

![Data Label Example](image)

When **DATALABELSPLITCHARDROP=TRUE**, the asterisks are removed from the label. When **DATALABELSPLITCHARDROP=FALSE**, each asterisk remains as the last character in the line prior to the new line.

**Default**
- **TRUE**. The split characters are dropped from the data label.

**Requirement**
- The **DATALABEL=** option and the **DATALABELSPLIT=TRUE** option must also be specified.

**Interaction**
- The **DATALABELSPLITCHAR=** option specifies the split characters.

**See**
- "boolean " on page 1479 for other Boolean values that you can use.

**DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT**

specifies the justification of the strings that are inside the data label blocks.

**AUTO**
- justifies the labels based on the **DATALABELPOSITION=** option.

<table>
<thead>
<tr>
<th><strong>DATALABELPOSITION=</strong> Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
</tbody>
</table>
**DATALABELPOSITION**= Value | Justification
---|---
TOP, CENTER, or BOTTOM | CENTER

**CENTER**
**LEFT**
**RIGHT**
justifies the labels center, left, or right, as specified.

The following figure shows an example in which **DATALABELPOSITION**=TOP.

**Note:** The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.

<table>
<thead>
<tr>
<th>AUTO</th>
<th>CENTER</th>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Group A</td>
<td>Product Group A</td>
<td>Product Group A</td>
<td>Product Group A</td>
</tr>
</tbody>
</table>

In this case, because **DATALABELPOSITION**=TOP, **AUTO** centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if **DATALABELPOSITION**=TOP, then the bottom center of the text box is positioned at the top of the marker.

**Default** | **AUTO**

**Requirement** | The **DATALABEL**= option and the **DATALABELSPLIT**=TRUE option must also be specified.

**Interaction** | This option has no effect if **DATALABELSPLIT**=FALSE, or if **DATALABELSPLIT**=TRUE and **DATALABELPOSITION**=AUTO.

**DATASKIN**=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the plot markers. The following figure shows large CIRCLEFILLED markers with each of the skins applied.
The **DATASKIN=** option value that is specified in the **BEGINGRAPH** statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

**Restriction**

Starting with **SAS 9.4M1**, the maximum number of skinned graphical elements is limited to 200 per plot in an **OVERLAY** or **PROTOTYPE** layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the **DATASKINMAX=** option in your ODS GRAPHICS statement to increase the maximum limit.

**Interactions**

This option overrides the **BEGINGRAPH** statement **DATASKIN=** option.

When a data skin is applied, all marker outlines are set by the skin, and the outline settings from the ODS style or from the **MARKERATTRS=** option are ignored.

**DATATRANSPARENCY=** *number*

specifies the degree of the transparency of the markers and data labels.

**Default**

0

**Range**

0–1, where 0 is opaque and 1 is entirely transparent

**Note**

This option does not apply to the **HISTOGRAM**, **DENSITY**, and **NORMAL** plots in the scatter-plot matrix.

**DIAGONAL=LABEL | (graph-list)**

specifies whether the diagonal cells of the matrix are labeled with the labels (names) of the required arguments, or with a plot. The plot for each diagonal cell consists of an overlay combination of a histogram, normal, or kernel curves.

**LABEL**

labels the diagonal cells of the matrix with the names of the required arguments.

**(graph-list)**

specifies the plots that are displayed in the diagonal cells of the matrix.

The **graph-list** is a space-separated list of one or more of the following:

**HISTOGRAM**

specifies a histogram.

**NORMAL**

specifies a normal density curve.

**KERNEL**

specifies a kernel density estimate.

**Default**

LABEL. Column labels (or names) are displayed in the diagonal cells.

**Requirement**

When specifying multiple graphs in the **graph-list**, you must separate the values with a space. For example, the following specification requests both a histogram and a normal density curve in each diagonal cell:
**INTERACTIONS**

The computation for HISTOGRAM, NORMAL, and KERNEL is always computed on all the data for the current column (including the FREQ= column, if used). The GROUP= option is not considered in any of these computations.

This option is ignored if the ROWVARS= option is used.

**NOTE**

When this option is specified, the labels are drawn around the outside of the matrix, and the matrix axes are dropped.

**ELLIPSE=(<ellipse-suboptions>)**

specifies that a confidence ellipse be included in each cell containing a scatter plot. The ellipse is always drawn behind the scatter points. *Ellipse-suboptions* can include one or more of the following:

**ALPHA=positive-number**

specifies the confidence level to compute for each ellipse.

Default 0.05

Range \(0 < \text{number} < 1\)

Tip ALPHA=0.05 represents a 95% confidence level.

**CLIP=TRUE | FALSE**

specifies whether the X and Y values for the ellipse are considered when determining the data ranges for the axes.

TRUE

the X and Y values for the ellipses are ignored when the axis ranges are determined.

Clipping occurs if the X and Y values for an ellipse exceed the axis range.

FALSE

the X and Y values for the ellipses contribute to the data range for each axis.

If necessary, each axis is extended in order to display the entire ellipse.

Default FALSE

See “boolean” on page 1479 for other Boolean values that you can use.

**TYPE=MEAN | PREDICTED**

specifies the type of ellipse.

MEAN

specifies a confidence ellipse of the mean

PREDICTED

specifies a prediction ellipse of the data

Default MEAN

See “ELLIPSE Statement” on page 472 for statistical details about how the ellipse is calculated.
### FREQ= numeric-column | expression

specifies a numeric column that provides frequencies for each observation that is read.

**Default**  
All observations have a frequency count of 1.

**Restriction**  
If the value of the numeric-column is missing or is less than 1, then the observation is not used in the analysis. If the value is not an integer, then only the integer portion is used.

**Note**  
If \( n \) is the value of the numeric column for a given observation, then that observation is used \( n \) times for the purposes of any statistical computation.

### GROUP= column | discrete-attr-var | expression

creates a distinct set of scatter markers, error bars, and data labels for each unique group value of the specified column.

**discrete-attr-var**  
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction**  
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

**Default**  
Each distinct group value might be represented in the graph by a different combination of color and marker symbol. Markers vary according to the ContrastColor and MarkerSymbol attributes of the GraphData1–GraphDataN and GraphMissing style elements.

**Interactions**  
The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of markers and colors.

The marker size is set by the MARKERATTRS= option.

The MARKERCHARACTER=, COLORRESPONSE=, and MARKERCOLORGRADIENT= options override the group settings for the marker symbol and marker color.
The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

**Note**  
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tip**  
The representations that are used to identify the groups can be overridden. For example, each distinct group value is represented by a different marker symbol, but the MARKERATTRS=(SYMBOL=marker) option could be used to assign the same symbol to all of the plot’s marker symbols, letting marker color indicate group values. Likewise, MARKERATTRS=(COLOR=color) could be used to assign the same color to all markers, letting marker symbol indicate group values.

**See**  
“DISCRETEATTRVAR Statement” on page 1435

### INCLUDEMISSINGGROUP=TRUE | FALSE

specifies whether missing values of the group variable are included in the plot.

**Default**  
TRUE

**Interaction**  
For this option to take effect, the GROUP= option must also be specified.

**Tip**  
The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**  
“boolean” on page 1479 for other Boolean values that you can use.

### INDEX=positive-integer-column | expression

specifies indices for mapping marker attributes (color and symbol) to one of the GraphData1–GraphDataN style elements.

**Requirements**  
The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interactions**  
For this option to take effect, the GROUP= option must also be specified.
The MARKERCHARACTER=, COLORRESPONSE=, and MARKERCOLORGRADIENT= options override the group settings for the marker symbol and marker color.

Notes

The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

Tip

You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

INSET=(info-options)

specifies what information is displayed in an inset. Insets appear in all cells of the matrix except the diagonal and are displayed as a small table of name-value pairs. The following info-options are available:

**NOBS**

displays the total number of observations where both the X and Y columns have nonmissing values.

If the FREQ= option is used, then this number is adjusted accordingly. The value of NOBS can be further adjusted by the use of the NOMISS=, WEIGHT=, and EXCLNPWGT= suboptions of the CORROPTS= option.

**PEARSON**

displays the Pearson product-moment correlation.

The computation of the correlation is affected by the FREQ= and CORROPTS= options. The computation is not done on a per group value when GROUP= is used.

**PEARSONPVAL**

displays the probability value for the Pearson product-moment correlation.

Tip

The location and appearance of the inset is controlled by the INSETOPTS= option.

See

PROC CORR in the documentation for Base SAS for statistical and computational details of these options.

Example

Here is an example of a typical inset:

```
N   150
r  0.96287
p(r)  <.0001
```

In this example, NOBS is represented by N, PEARSON is represented by r, and PEARSONPVAL is represented by p(r).

INSETOPTS=(appearance-options)

specifies location and appearance options for the inset information. The appearance options can be any one or more of the settings that follow. The options must be enclosed in parentheses, and each option is specified as a name=value pair.
AUTOALIGN=NONE | AUTO | *(location-list)*
specifies whether the inset is automatically aligned within the layout.

**NONE**
do not automatically align the inset.

The inset’s position is therefore set by the HALIGN= and VALIGN= **appearance-options**.

**AUTO**

attempts to center the inset in the area that is farthest from any surrounding markers.

Data cells might have different inset placements.

*(location-list)*

restricts the inset’s possible locations to those locations in the specified **location-list**, and uses the **location-list** position that least collides with the data cell’s other graphics features.

The **location-list** is space-separated and can contain any of these locations: TOPLEFT TOP TOPRIGHT LEFT CENTER RIGHT BOTTOMLEFT BOTTOM BOTTOMRIGHT. Example:

AUTOALIGN=*(TOPRIGHT TOPLEFT)*

**Default** NONE

**Interaction** When AUTOALIGN=AUTO or *(location-list)*, the enclosing layout statement’s HALIGN= and VALIGN= **appearance-options** are ignored.

**BACKGROUNDCOLOR=** **style-reference** | **color**

specifies the color of the inset background

**style-reference**

specifies a style reference in the form **style-element** : **style-attribute**. Only the style-attribute named COLOR or CONTRASTCOLOR is used.

**Default** GraphWalls:Color style reference

**BORDER=** TRUE | FALSE

specifies whether a border is displayed around the inset.

**Default** FALSE

**See** “boolean ” on page 1479 for other Boolean values that you can use.

**HALIGN=** LEFT | CENTER | RIGHT

specifies the horizontal alignment of the inset.

**Default** LEFT

**Interaction** This option is ignored when AUTOALIGN= is not NONE and the parent layout is an overlay-type layout.

**OPAQUE=** TRUE | FALSE

specifies whether the inset background is opaque (TRUE) or transparent (FALSE).

**Default** FALSE
Interaction  When this option is set to FALSE, the background color is not used.

See  "boolean " on page 1479 for other Boolean values that you can use.

TEXTATTRS=style-element | style-element (text-options) | (text-options)
specifies the text properties of the entire inset.

Default  The GraphDataText style element.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

TITLE="string"
specifies a title for the inset.
The title is added at the top of the inset and spans the full inset width.

Note  Space is not reserved for the title when this option is not set

Tip  Text properties for the title string can be set with TITLEATTRS=.

TITLEATTRS=style-element | style-element (text-options) | (text-options)
specifies the text properties of the inset’s title string.

Default  The GraphValueText style element.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

VALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment of the inset.

Default  TOP

Interaction  This option is ignored when AUTOALIGN= is not NONE and the parent layout is an overlay-type layout.

LABELSTRIP=TRUE | FALSE
specifies whether leading and trailing blanks are stripped from marker characters or data labels that have a fixed position before they are displayed in the plot. The MARKERCHARACTER= option specifies the column that provides the marker strings that are to be used in place of marker symbols.

Default  FALSE

Interactions  This option effects marker strings only when the MARKERCHARACTER= option is specified.

This option effects data labels only when DATALABEL= is specified and DATALABELPOSITION= is not AUTO.
Tip

Stripping the blanks from the numeric value strings helps center each string relative to its data point. Stripping is useful when you want to overlay the data values near or inside the markers for a plot.

See

“boolean” on page 1479 for other Boolean values that you can use.

MARKERATTRS=style-element | style-element (marker-options) | (marker-options)

specifies the attributes of the data markers.

Defaults

For non-grouped data, GraphDataDefault style element.

For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

Interactions

If the COLORRESPONSE= or MARKERCOLORGRADIENT= option is specified, then this option’s COLOR= suboption is ignored.

If the MARKERCHARACTER= option is specified, then this option’s SYMBOL= and WEIGHT= suboptions are ignored.

This option’s COLOR= suboption overrides the default behavior for grouped data. When the COLOR= suboption is specified in that case, all markers have the same color, and the marker symbol alone distinguishes the markers.

This option’s SYMBOL= suboption overrides the default behavior for grouped data. When the SYMBOL= suboption is specified in that case, all markers have the same symbol, and the symbol color alone distinguishes the markers.

The TRANSPARENCY= fill option overrides this option’s DATATRANSPARENCY= suboption.

Note

When style-element is specified, only the style element’s MARKERSYMBOL, CONTRASTCOLOR, and MARKERSIZE attributes are used.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Marker Options” on page 1523 for available marker-options.

MARKERCHARACTER=column | expression

specifies a column that defines strings that are to be used instead of marker symbols.
Each string is centered horizontally and vertically at the data point. The data point positions are not adjusted to prevent text overlap.

**Interactions**
This option overrides the DATALABEL= option.

If the GROUP= option is also used, then color is displayed for a DISCRETE legend, but the character is not displayed in the legend. For text strings, the same colors are applied to the text strings as would have been applied to markers.

**Note**
If a numeric column is used, then its values are converted to strings using the format associated with the column or using BEST6 if no format is defined.

**Tips**
Lengthy strings might be clipped by the plot border. To reduce clipping, you can use the OFFSETMIN= and OFFSETMAX= suboptions of the XAXISOPTS= and YAXISOPTS= options to increase the axis offsets.

You can use the MARKERCHARACTERPOSITION= option to change the justification of the marker character.

You can use the LABELSTRIP= option to strip the leading and trailing blanks from numeric value strings in order to center each string on its data point.

**MARKERCHARACTERATTRS=**

specifies the color and font attributes of the marker character specified on the MARKERCHARACTER= option.

**Defaults**
For non-grouped data, the GraphDataText style element.

For grouped data, GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

**Interactions**
For this option to take effect, the MARKERCHARACTER= option must also be used.

When the GROUP= option is also specified, each distinct group value might be represented by a different color (depending on the ODS style setting or the setting on the INDEX= option). The marker
character that is associated with the group is assigned the group color.

This option's COLOR= suboption can be used to specify a single color for all marker characters in a grouped plot, without affecting items that have a group color, such as error bars and marker symbols.

See "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Text Options" on page 1525 for available text-options.

**MARKERCHARACTERPOSITION=** CENTER | TOP | BOTTOM | LEFT | RIGHT | TOPLEFT | TOPRIGHT | BOTTOMLEFT | BOTTOMRIGHT

specifies the justification of the marker characters. The following figure shows the effect of each of the values on the position of marker character M.

![Marker Character Position Examples](image)

**Note:** The red dots represent the center in each case for reference.

**Default** CENTER

**Interaction** This option is ignored if the MARKERCHARACTER= option is not specified.

**Tip** You can use the LABELSTRIP= option to strip the leading and trailing blanks from value strings in order to properly justify each string on its data point.

**MARKERCOLORGRADIENT=** numeric-column | range-attr-var | expression

in SAS 9.4M1 and earlier releases, specifies the column or range attribute map variable that is used to determine the marker colors.

**Note:** Starting with SAS 9.4M2, the MARKERCOLORGRADIENT= option is deprecated and replaced with the COLORRESPONSE= option. The syntax and functionality are the same. The MARKERCOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is preferred.

**Note** Starting with SAS 9.4M2, if you use the MARKERCOLORGRADIENT= option, then be aware that the TIP=, TIPFORMAT=, and TIPLABEL= options recognize the MARKERCOLORGRADIENT role and not the COLORRESPONSE role.
COLORRESPONSE= on page 799

**MATRIXTYPE=FULL | UPPERTRIANGLE | LOWERTRIANGLE**

specifies whether to display the full matrix, or just the upper or lower triangle of the matrix. By default, the full matrix is displayed. The cells in the grid are filled beginning with the cell that is specified by the START= option. When you display only the upper or lower triangle, you can use the START= option to control the orientation of the triangle. The following figure shows the effect of the MATRIXTYPE= option when START=TOPLEFT and DIAGONAL=(HISTOGRAM).

The next figure shows the effect of the START=BOTTOMLEFT option on the previous graph.

**DEFAULT**

This option is ignored if the ROWVARS= option is used.

The **START=** option specifies the corner where the matrix fill begins.

The **DIAGONAL=** option specifies the content of the diagonal cells.

**NAME="string"**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.
Restrictions  The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**REVERSECOLORMODEL=TRUE | FALSE**

specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

Default  FALSE

See  COLORMODEL=

“boolean " on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

(role-name-list)  a space-separated list of role-name = column pairs.

Example  The following example assigns the column Obs to the user-defined role TIP:

```
ROLENAME=(TIP1=OBS)
```

Default  No user-defined roles

Requirement  The role names that you choose must be unique and different from the predefined roles X, Y, DATALABEL, MARKERCHARACTER, COLORRESPONSE or MARKERCOLORGRADIENT, GROUP, and INDEX.

**ROWVARS=(numeric-column-list)**

specifies a secondary list of numeric columns to be paired with the required column list that is specified by the SCATTERPLOTMATRIX statement. The labels for the columns appear vertically on the left side of the matrix.

Requirement  All of the columns must be numeric.

Interaction  When this option is specified, the DIAGONAL= and MATRIXTYPE= options are ignored.

**START=TOPLEFT | BOTTOMLEFT**

specifies whether to start populating the matrix from the top left or bottom left corner.

Default  TOPLEFT

**SUBPIXEL=AUTO | OFF**

specifies whether subpixel rendering is used for image output when the scatter plots are rendered.

Note:  This feature applies to SAS 9.4M3 and to later releases.
<table>
<thead>
<tr>
<th>AUTO</th>
<th>sets subpixel rendering OFF, unless SUBPIXEL=ON is specified in the BEGINGRAPH statement or in an ODS GRAPHICS statement. In that case, subpixel rendering is ON.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>disables subpixel rendering for image output only for the scatter plots in this scatter-plot matrix.</td>
</tr>
</tbody>
</table>

**Default**  
AUTO

**Restriction**  
This option applies only to the scatter plots in the matrix. It does not affect the confidence ellipses.

**Requirement**  
Anti-aliasing must be enabled for this option to have any effect.

**Notes**  
This option affects subpixel rendering for the scatter plots only in this scatter-plot matrix. Subpixel rendering for other plots in the graph is not affected.

For vector-graphics output, this option is ignored, and subpixel rendering is always enabled.

When subpixel rendering is used for the graph but is turned OFF for this scatter-plot matrix, some elements in the scatter plots such as the plot markers might be offset a half pixel, which can make them appear blurry in the image output.

**Tips**  
Anti-aliasing is enabled by default. If anti-aliasing has been disabled, use the ANTIALIAS=ON option in the ODS GRAPHICS statement to re-enable it.

Anti-aliasing is disabled automatically for this plot when the resources required to anti-alias it exceed a preset threshold. When anti-aliasing is disabled for this or any other plot in the graph, subpixel rendering is disabled for the entire graph. A note is written to the SAS log that provides information about how to use the ANTIALIASMAX= option in an ODS GRAPHICS statement to re-enable anti-aliasing.

To disable subpixel rendering for the scatter plots and the ellipses, specify SUBPIXEL=OFF in the template’s BEGINGRAPH statement or in an ODS GRAPHICS statement.

**See**  
“Using Subpixel Rendering” in SAS Graph Template Language: User’s Guide

“ODS GRAPHICS Statement” on page 1483 for information about the ANTIALIAS=, ANTIALIASMAX=, and SUBPIXEL= options.

**TIP=(role-list) | NONE**  
specifies the information to display when the cursor is positioned over the scatter points. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the scatter plot can be specified along with roles that do.

**(role-list)**  
an ordered, space-separated list of unique SCATTERPLOTMATRIX and user-defined roles.
SCATTERPLOT roles include: X, Y, GROUP, DATALABEL, MARKERCHARACTER, and COLORRESPONSE or MARKERCOLORGRADIENT.

User-defined roles are defined with the ROLENAMESPACE option.

Requirement If you use the COLORRESPONSE= option, you must specify the COLORRESPONSE role for the color values. Likewise, if you use the MARKERCOLORGRADIENT= option, you must specify the MARKERCOLORGRADIENT role for the color values. Although they are functionally the same, you cannot mix the COLORRESPONSE= and MARKERCOLORGRADIENT= options, and their corresponding roles.

Example The following example displays data tips for the columns assigned to the roles TIP1, TIP2, TIP3, and TIP4:

```
ROLENAME=(TIP1=ID TIP2=AGE TIP3=HEIGHT TIP4=WEIGHT)
TIP=(TIP1 TIP2 TIP3 TIP4)
```

NONE suppresses data tips from the plot.

Default The columns assigned to these roles are automatically included in the data tip information: current X, current Y, DATALABEL, MARKERCHARACTER, COLORGROUP or MARKERCOLORGRADIENT, and GROUP.

Requirement To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)

specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example

```
ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
```

Default The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction Only the roles that appear in the TIP= option are used.

Requirement A column must be assigned to each of the specified roles. (See the ROLENAMESPACE= option.)
**TIPLABEL=(role-label-list)**
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

*role-label-list*

a space-separated list of `rolename ="string"` pairs.

**Example**

```plaintext
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

**Default**
The column label or column name of the column assigned to the role.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**WALLCOLOR=style-reference | color**
specifies the fill color of the plot wall area.

*style-reference*

specifies a style reference in the form `style-element:style-attribute`.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

**Default**
The GraphWalls:Color style reference.

**Interaction**
This option is ignored if WALLDISPLAY=NONE or WALLDISPLAY=(OUTLINE).

**WALLDISPLAY=STANDARD | ALL | NONE | (display-options)**
specifies whether the plot’s wall and wall outline are displayed.

**STANDARD**
displays a filled wall.

The setting of the FRAMEBORDER=ON | OFF attribute of the GraphWalls style element determines whether the wall outline is displayed.

**ALL**
displays a filled, outlined wall.

**NONE**
displays no wall, no wall outline.

*(display-options)*
a space-separated list of one or more display options enclosed in parentheses.

The list can include one or both of the following options:

**OUTLINE**
displays the wall outline.

**FILL**
displays a filled wall area.

**Default**
STANDARD

**Tips**
Use the WALLCOLOR= option to control the fill color of the wall.
The appearance attributes of the wall fill and outline are set by the GraphWalls style element.

Details

By default, the SCATTERPLOTMATRIX statement produces a symmetric scatter plot matrix. For \( n \) columns, it produces an \( n \) columns by \( n \) rows matrix of scatter plots. By default, the columns of the matrix are in the same left-to-right order as the order of the numeric-column-list. The rows of the matrix are in the same bottom-to-top order as the numeric-column-list. You can reverse the direction of the diagonal by setting START=TOPLEFT.

To produce a rectangular matrix of scatter plots, use the ROWVARS= option. Specifying \( n \) columns in the SCATTERPLOTMATRIX statement and \( m \) columns on the ROWVARS= option produces an \( n \)-columns by \( m \)-rows matrix of scatter plots. For example, the following statement specifies 2 columns on SCATTERPLOTMATRIX and 3 columns on the ROWVARS= option to produce the 2-columns by 3-rows matrix:

```
SCATTERPLOTMATRIX Height Weight / ROWVARS=(Age Height Weight);
```

The SCATTERPLOTMATRIX statement cannot appear within an overlay-type layout. It generates its own matrix of plots and is typically placed in a LAYOUT GRIDDED block.

If there are missing values in a column or a row, then all of the points that can be plotted are plotted in each scatter plot.

Example: SCATTERPLOTMATRIX Statement

The following graph was generated by the “Example Program” on page 822:
Example Program

```sas
proc template;
  define statgraph scatterplotmatrix;
  begingraph;
    entrytitle "Scatter Plot Matrix";
    layout gridded;
      scatterplotmatrix
        sepalwidth sepallength petalwidth petallength /
          group=species name="matrix"
        ;
      discretelegend "matrix"
    ;
  endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.iris template=scatterplotmatrix;
run;
```

SERIESPLOT Statement

Displays a series of line segments that connect observations of input data.

**Tip:** Starting with **SAS 9.4M3**, subpixel rendering is enabled by default. To disable subpixel rendering, specify **SUBPIXEL=OFF** in the **BEGINGRAPH** statement or in an **ODS GRAPHICS** statement. For information about the **BEGINGRAPH** statement **SUBPIXEL=** option, see **SUBPIXEL=** on page 35. For information about the **ODS GRAPHICS** statement **SUBPIXEL=** option, see “**ODS GRAPHICS Statement**” on page 1483.

**Syntax**

```
SERIESPLOT X=column | expression
Y=column | expression </options>;
```

**Summary of Optional Arguments**

**Appearance options**

- **ARROWHEADDIRECTION=** specifies a position for arrowheads.
  - **NONE** | **START** | **END** | **BOTH** | **column**

- **ARROWHEADSCALE=** specifies an arrowhead scale factor based on the thickness of the arrow line.
  - **positive-number** | **numeric-column** | **expression**

- **ARROWHEADDIRECTION=** specifies a shape for arrowheads.
  - **OPEN** | **FILLED** | **BARBED** | **column**

- **BREAK=** breaks the plot line at missing values of the Y variable.
  - **TRUE** | **FALSE**

- **CLUSTERWIDTH=** specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group
clusters as a fraction of the minimum interval between adjacent data values.

COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the column or range attribute variable to use to map the line and marker colors.

DATASKIN=None | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the series lines.

DATATRANSPARENCY=number
specifies the degree of the transparency of the line, markers, line label, and data labels, when displayed.

DISPLAY=(display-options) | STANDARD | ALL
specifies which series-line features to display.

ERRORBARATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the error bars that are associated with the data points.

ERRORBARCAPSCALE=positive-number
specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

ERRORBARCAPSHAPE=SERIF | NONE
specifies whether the error bars have a serif cap.

FILLEDOUTLINEDMARKERS=TRUE | FALSE
specifies whether markers are drawn with both fill and an outline.

INDEX=positive-integer-column | expression
specifies indices for mapping line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the series line.

LINECOLORGROUP=column | discrete-attr-var | expression
specifies a column that determines the line colors for a grouped plot independently of the GROUP= column.

LINEJOIN=MITER | ROUND | BEVEL
specifies how to shape the vertices of the series line.

LINEPATTERNGROUP=column | discrete-attr-var | expression
specifies a column that determines the line patterns for a grouped plot independently of the GROUP= column.

LINETHICKNESSMAX=dimension
specifies the maximum line thickness when a response variable is used to determine the line thickness.

LINETHICKNESSMAXRESPONSE=numeric | scalar-numeric-expression
specifies the response value that corresponds to the maximum line thickness.

LINETHICKNESSMIN=dimension
specifies the minimum line thickness when a response variable is used to determine the line thickness.

LINETHICKNESSRESPONSE=numeric-column | expression
specifies a response column or range attribute variable that is used to map a line thickness to each group value.

MARKERATTRS=style-element | style-element (marker-options) | (marker-options)
specifies the attributes of the data markers.
MARKERCOLORGROUP=column | discrete-attr-var | expression
specifies a column that determines the marker colors for a grouped plot independently of the GROUP= column.

MARKERFILLATTRS=style-element | (fill-options)
specifies the appearance of the filled markers.

MARKEROUTLINEATTRS=style-element | (line-options)
specifies the appearance of the marker outlines.

MARKERSYMBOLGROUP=column | discrete-attr-var | expression
specifies a column that determines the marker symbols for a grouped plot independently of the GROUP= column.

SMOOTHCONNECT=TRUE | FALSE
specifies that a smoothed line passes through all vertices.

SPLINEPOINTS=positive-integer
specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.

SPLINETYPE=NONE | QUADRATICBEZIER
specifies the type of spline interpolation that is used to draw the series line.

XERRORLOWER=numeric-column | expression
specifies values for the lower endpoints on the X error bars. The error bars are drawn from the markers to the endpoints.

XERRORUPPER=numeric-column | expression
specifies values for the upper endpoints on the X error bars.

YERRORLOWER=numeric-column | expression
specifies values for the lower endpoints on the Y error bars.

YERRORUPPER=numeric-column | expression
specifies values for the upper endpoints on the Y error bars.

Axes options

CLUSTERAXIS=AUTO | X | Y
specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Connect options

CONNECTORDER=XVALUES | XAXIS
specifies how to connect the data points to form the series line.

Data tip options

ROLENAMESPACE=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over the series line.

TIPFORMAT=(role-format-list)
  specifies display formats for tip columns.

TIPLABEL=(role-label-list)
  specifies display labels for tip columns.

**Label options**

**CURVELABEL=**"string" | column | expression
  specifies a label for the series line.

**CURVELABELATTRS=**style-element | style-element (text-options) | (text-options)
  specifies the color and font attributes of the series-line labels.

**CURVELABELLOCATION=**INSIDE | OUTSIDE
  specifies the location of the series-line label relative to the plot area.

**CURVELABELPOSITION=**AUTO | MAX | MIN | START | END
  specifies the position of the series-line label relative to the series line.

**CURVELABELSPLIT=**TRUE | FALSE
  specifies whether to split the series-line label at the specified split characters.

**CURVELABELSPLITCHAR=**"character-list"
  specifies one or more characters on which the series-line label can be split if needed.

**CURVELABELSPLITCHARDROP=**TRUE | FALSE
  specifies whether the split characters are included in the series-line label text.

**CURVELABELSPLITJUSTIFY=**AUTO | CENTER | LEFT | RIGHT
  specifies the justification of the strings that are inside the series-line label block.

**DATALABEL=**column | expression
  specifies a column that supplies values for the data point labels.

**DATALABELATTRS=**style-element | style-element (text-options) | (text-options)
  specifies the color and font attributes of the data labels.

**DATALABELPOSITION=**AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
  specifies the location of the data labels relative to the vertices of the series line and the markers, when displayed.

**DATALABELSPLIT=**TRUE | FALSE
  specifies whether to split the data labels at the specified split characters.

**DATALABELSPLITCHAR=**"character-list"
  specifies one or more characters on which the data labels can be split if needed.

**DATALABELSPLITCHARDROP=**TRUE | FALSE
  specifies whether the split characters are included in the data labels.

**DATALABELSPLITJUSTIFY=**AUTO | CENTER | LEFT | RIGHT
  specifies the justification of the strings that are inside the data label blocks.

**LEGENDLABEL=**"string"
  specifies a label to be used in a discrete legend for this plot.

**Midpoint options**

**DISCRETEOFFSET=**number
specifies an amount to offset all series lines and markers from discrete X values, or discrete Y values, or both.

GROUP=column | discrete-attr-var | expression
creates a separate series plot for each unique group value in the specified column.

GROUPDISPLAY=OVERLAY | CLUSTER
specifies how marker groups are positioned for the coordinate pairs.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

ODS options

URL=string-column
specifies an HTML page to display when a point or a segment of the curve is selected.

Plot reference options

NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments

X=column | expression
specifies the column for the X values.

Y=column | expression
specifies the column for the Y values.

Optional Arguments

ARROWHEADPOSITION=NONE | START | END | BOTH | column
specifies a position for arrowheads.

Note: This feature applies to SAS 9.4M3 and to later releases.

NONE
no arrowheads are displayed.

START
an arrowhead is displayed at the starting point of each line.

END
an arrowhead is displayed at the ending point of each line.

BOTH
an arrowhead is displayed at each end of each line.

column
specifies a column that provides an arrowhead position for each group value.

Default
NONE

Restriction
When you specify a column and the GROUP= option is in effect, the arrowhead position values are assumed to be constant for each
group value. If the column has multiple values for a single group value, then only one of the arrowhead position values is used for that group.

See

“Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

ARROWHEADSCALE= positive-number | numeric-column | expression

specifies an arrowhead scale factor based on the thickness of the arrow line.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default 1.0

Restriction The arrowhead scale values are assumed to be constant for each line. If a column or expression provides multiple values for a line, only one of the values is used.

Interaction This option is ignored when ARROWHEADPOSITION=NONE is in effect.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

See “Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

ARROWHEADSHAPE= OPEN | FILLED | BARBED | column

specifies a shape for arrowheads.

Note: This feature applies to SAS 9.4M3 and to later releases.

The following figure shows each of the arrowhead shapes.

The following figure shows each of the arrowhead shapes.

\[
\begin{array}{c}
\text{OPEN} \\
\text{FILLED} \\
\text{BARBED}
\end{array}
\]

Default OPEN

Restriction When you specify a column and the GROUP= option is in effect, the arrowhead shape values are assumed to be constant for each group value. If the column has multiple values for a single group value, only one of the arrowhead shape values is used for that group.

Interaction This option is ignored when ARROWHEADPOSITION=NONE is in effect.

See “Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

BREAK= TRUE | FALSE

breaks the plot line at missing values of the Y variable.

Default FALSE
Interaction

This option is ignored when the Y axis is discrete and option INCLUDEMISSINGDISCRETE=TRUE is specified in the BEGINGRAPH statement. To break the line in that case, remove INCLUDEMISSINGDISCRETE=TRUE from the BEGINGRAPH statement.

Note

When this option is set to FALSE, missing values are skipped and a continuous line is drawn.

See

“boolean ” on page 1479 for other Boolean values that you can use.

CLUSTERAXIS= AUTO | X | Y

specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

AUTO
uses the discrete axis for clustering groups when only one axis is discrete. Uses the X axis for clustering if both axes are discrete or interval.

X    Y
uses the X or Y axis for clustering groups.

Default AUTO

Interaction

The GROUPDISPLAY= option must be set to CLUSTER for this option to have any effect.

CLUSTERWIDTH= number

on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

Default 0.85

Range 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Interactions

For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

When markers are displayed for interval data and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

COLORMODEL= color-ramp-style-element | (color-list)

specifies a color ramp to use with the COLORRESPONSE= option.

color-ramp-style-element
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

STARTCOLOR
specifies the color for the smallest data value of the COLORRESPONSE= column.
NEUTRALCOLOR
   specifies the color for the midpoint of the range of the
   COLORRESPONSE= column.

ENDCOLOR
   specifies the color for the highest data value of the COLORRESPONSE= column.

(color-list)
   specifies a space-separated list of colors to use in the color ramp. You can
   use style attribute references such as GraphData3:Color, color names, or
   RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list
   can contain a mix of style attribute references, color names, and color codes.

Requirement  The list of colors must be enclosed in parentheses.

See    “color ” on page 1480

Default   The ThreeColorAltRamp style element.

Interaction  For this option to take effect, the COLORRESPONSE= option must
             also be specified.

COLORRESPONSE= numeric-column | range-attr-var | expression
   specifies the column or range attribute variable to use to map the line and
   marker colors.

Note:  This feature applies to SAS 9.4M3 and to later releases.

range-attr-var
   specifies a range attribute map variable that is defined in a RANGEATTRVAR
   statement.

   “RANGEATTRVAR Statement”

Restriction  A range attribute map variable specification must be a direct
             reference to the attribute map variable. It cannot be set as a
             dynamic variable.

See    “RANGEATTRVAR Statement” on page 1448

When a numeric column or expression is specified, the range of column or
expression values are linearly mapped to the color ramp that is specified by the
COLORMODEL= option. When a range attribute map variable is specified, the
colors that are defined in the associated range attribute map are used instead.

Requirement  The COLORRESPONSE values should remain constant for each
             group value in a grouped plot and for the entire plot in an
             ungrouped plot. If the COLORRESPONSE column has multiple
             values for a single GROUP value or for a non-grouped plot,
             unexpected results might occur.

Interaction  When this option is specified without the GROUP= option, only a
             single line is generated for the plot, and the line color is derived
             from the COLORRESPONSE= value.

              When the GROUP= option, LINECOLORGROUP= option, and
              MARKERCOLORGROUP= options are used with the
COLORRESPONSE= option, the COLORRESPONSE= option controls the color attributes.

Suboption COLOR= in the DATALABELATTRS= option overrides this option for the data label color attribute.

This option overrides suboption COLOR= in the LINEATTRS= option and varies the line color according to the color gradient or the attribute map.

Suboption COLOR= in the MARKERATTRS= option or in the FILLEDOUTLINEDMARKERS= option overrides this option for the marker colors.

Tips
To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

For a numeric column or expression, the ThreeColorAltRamp style element defines the line color gradient.

CONNECTORDER=XVALUES | XAXIS
specifies how to connect the data points to form the series line.

XVALUES
connects data points in the order read from the X column.

Note
In SAS 9.4M6 and earlier releases, unexpected results might occur when XVALUES is specified and data stored in a CAS in-memory table is used. In that case, use XAXIS instead. Starting with SAS Viya 3.5 and SAS Studio 5.2, XVALUES is ignored when data stored in a CAS in-memory table is used.

XAXIS
connects data points as they occur min-to-max along the X axis.

Defaults
XVALUES if the data is a SAS data set
XAXIS if the data is a CAS in-memory table

Tip
For certain types of series lines (for example, time series) when the input data might not be sorted by the X column, set this option to XAXIS to assure the expected connect order.

CURVELABEL= "string" | column | expression
specifies a label for the series line.

Default
No series-line label is displayed

Restrictions
When the GROUP= option is specified, "string" and expression are not valid. Use column in that case.

When the GROUP= option is not specified, column is not valid. Use "string" or expression in that case.

The line label for missing values is ignored.

Tip
The font and color attributes for the label are specified by the CURVELABELATTRS= option.
CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the series-line labels. When the GROUP= option is used, each distinct group value might be represented by a different color. The series-line label that is associated with the group is assigned the group color. This option can be used to specify a single color for all series-line labels in a plot, without affecting items that have the group color, such as lines and marker symbols.

Defaults
For non-grouped data, the GraphValueText style element.

For grouped data, text color is derived from the GraphData1:ContrastColor–GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

Interactions
For this option to take effect, the CURVELABEL= option must also be used.

This option’s COLOR= setting overrides the colors indicated by the GROUP= option.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

CURVELABELLOCATION=INSIDE | OUTSIDE
specifies the location of the series-line label relative to the plot area.

INSIDE
locates the labels inside the plot area

OUTSIDE
locates the labels outside the plot area

Default
INSIDE

Restriction
OUTSIDE cannot be used when the SERIESPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes might be external to the grid.

Interactions
For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELPOSITION= option to determine where the series-line labels appear. For more information, see “Location and Position of Curve Labels” on page 200.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
specifies the position of the series-line label relative to the series line.

AUTO
automatically locates the series-line label near the end series line along unused axes whenever possible (typically Y2 or X2) in order to avoid collision with tick values.
Restriction This option is used only when CURVELABELLOCATION=OUTSIDE.

MAX
forces the series-line label to appear near maximum series values (typically, to the right)

MIN
forces the series-line label to appear near minimum series values (typically, to the left)

START
forces the series-line label to appear near the beginning of the series line.

Restriction This option is used only when CURVELABELLOCATION=INSIDE.

Tip This option is particularly useful when the series line has a spiral shape.

END
forces the series-line label to appear near the end of the series line.

Restriction This option is used only when CURVELABELLOCATION=INSIDE.

Tip This option is particularly useful when the series line has a spiral shape.

Defaults AUTO when CURVELABELLOCATION=OUTSIDE.

END when CURVELABELLOCATION=INSIDE.

Restriction The AUTO setting is ignored if CURVELABELLOCATION=INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

Interactions For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELLOCATION= option to determine where the series-line label appears. For more information, see “Location and Position of Curve Labels” on page 200.

Note When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the series-line label might fall outside of the graph area. In that case, the series-line label might not be displayed or might be positioned incorrectly.

Tip When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.

CURVELABELSPLIT=TRUE | FALSE
specifies whether to split the series-line label at the specified split characters. When a series-line label is split, the label is split on each occurrence of the specified split characters.
**CURVELABELSPLITCHAR=**"character-list"

This option specifies one or more characters on which the series-line label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the series-line label. In that case, all of the specified split characters together are treated as a single split character.

When **CURVELABEL=** is specified and **CURVELABELSPLIT=**TRUE, the series-line label is split unconditionally at each occurrence of any of the specified split characters. If the series-line label does not contain any of the specified characters, then the label is not split.

**"character-list"**

- one or more characters with no delimiter between each character and enclosed in quotation marks.

**Default**

A blank space

**Requirements**

The list of characters must be enclosed in quotation marks.

- Multiple characters must be specified with no delimiters. For example, to specify the split characters a, b, and c, use the following option:

  ```
  curvelabelsplitchar="abc"
  ```

**Interactions**

This option has no effect if **CURVELABELPOSITION=**AUTO.

**Notes**

- When multiple characters are specified, the order of the characters in the list is not significant.
- The split characters are case sensitive.

**Tip**

Use the **CURVELABELSPLITJUSTIFY=** option to specify the justification of the strings in the series-line label block.

**CURVELABELSPLITCHARDROP=**TRUE | FALSE

This option specifies whether the split characters are included in the series-line label text.

- **TRUE**
  - drops the split characters from the series-line label text.
FALSE
includes the split characters in the series-line label text.

When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a series-line label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

Note: The horizontal line to the left of the label represents the maximum end of the series line for reference.

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default TRUE. The split characters are dropped from the series-line label.

Requirement The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction The CURVELABELSPLITCHAR= option specifies the split characters.

See “boolean ” on page 1479 for other Boolean values that you can use.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the series-line label block.

AUTO
justifies the labels based on the CURVELABELPOSITION= option.

<table>
<thead>
<tr>
<th>CURVELABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
</tbody>
</table>
CURVELABELPOSITION= Value  Justification

MIN or START  RIGHT

CENTER  
LEFT  
RIGHT  

justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the series line for reference.

<table>
<thead>
<tr>
<th>AUTO</th>
<th>CENTER</th>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Group A</td>
<td>Product Group A</td>
<td>Product Group A</td>
<td>Product Group A</td>
</tr>
</tbody>
</table>

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

Default  AUTO

Requirement  The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

Interaction  This option has no effect if CURVELABELPOSITION=AUTO.

DATALABEL=column | expression  

specifies a column that supplies values for the data point labels.

Default  No data labels are displayed

Note  The label positions are adjusted to prevent the labels from overlapping.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)  

specifies the color and font attributes of the data labels.

Defaults  For non-grouped data, the GraphDataText style element.

For grouped data, the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

Interactions  For this option to take effect, the DATALABEL= option must also be specified.

This option’s COLOR= setting overrides the colors indicated by the GROUP= option.
When the DATALABELPOSITION=AUTO option is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

When the GROUP= option is used, each distinct group value might be represented by a different color. The data label that is associated with the group is assigned the group color. This option can be used to specify a single color for all data labels in a plot, without affecting items that have the group color, such as error bars and marker symbols.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

DATALABELPOSITION=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT

specifies the location of the data labels relative to the vertices of the series line and the markers, when displayed.

AUTO

uses a data-label collision avoidance algorithm to position the labels.

See “LABELPLACEMENT=AUTO | GREEDY | SA” on page 33

Note When AUTO is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

TOPRIGHT

TOP

TOPLEFT

LEFT

CENTER

RIGHT

BOTTOMLEFT

BOTTOM

BOTTOMRIGHT

specifies a location for all of the data labels.

Default AUTO

DATALABELSPLIT=TRUE | FALSE

specifies whether to split the data labels at the specified split characters.

Default FALSE. The data labels are not split.

Requirement The DATALABEL= option must also be specified.

Interactions The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

This option has no effect when DATALABELPOSITION=AUTO.

See “boolean ” on page 1479 for other Boolean values that you can use.
**DATALABELSPLITCHAR=character-list**

specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

*character-list*

one or more characters with no space between each character and enclosed in quotation marks.

**Default**

A blank space

**Requirements**

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

datalabelsplitsplitchar="abc"

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interactions**

This option has no effect if DATALABELPOSITION=AUTO.

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels.

**TRUE**

drops the split characters from the data label.

**FALSE**

includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:

- the data label text for this label is Product*Group*A
DATALABELSPLIT=TRUE
DATALABELSPLITCHARDROP=TRUE | FALSE
DATALABELSPLITCHAR=***

When DATALABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When DATALABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default: TRUE. The split characters are dropped from the data label.

Requirement: The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interaction: The DATALABELSPLITCHAR= option specifies the split characters.

See: “boolean” on page 1479 for other Boolean values that you can use.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the data label blocks.

AUTO

justifies the labels based on the DATALABELPOSITION= option.

<table>
<thead>
<tr>
<th>DATALABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

CENTER
LEFT
RIGHT

justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.

Note: The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.
In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

**Default**

AUTO

**Requirement**
The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction**
This option has no effect if DATALABELPOSITION=AUTO.

**DATASKIN**

NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN

enhances the visual appearance of the series lines. The following figure shows a grouped series plot with each of the skins applied.

**Default**
The DATASKIN= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

**Restriction**
Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

**Interactions**
This option overrides the BEGINGRAPH statement DATASKIN= option.

The data skin appearance is based on the LINEATTRS= color.

**DATATRANSPARENCY**

Specifies the degree of the transparency of the line, markers, line label, and data labels, when displayed.
DISCRETEOFFSET=number
specifies an amount to offset all series lines and markers from discrete X values, or discrete Y values, or both. This option is useful when graphing multiple response variables side by side on a common axis.

Default 0 (no offset, all series lines and markers are centered on the discrete X values, or discrete Y values, or both)

Range 0–1, where 0 is opaque and 1 is entirely transparent

This option applies to discrete axes only. For nondiscrete axes, this option is ignored.

Tip Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

DISPLAY=(display-options) | STANDARD | ALL
specifies which series-line features to display.

(display-options)
a space-separated list of one or more options enclosed in parentheses.

Currently, only the MARKERS option is supported, which displays a series line with markers.

STANDARD displays a series line without markers.

ALL displays a series line with markers.

Default STANDARD

Tip Use the MARKERATTRS= and LINEATTRS= options to control the appearance of the line and markers.

ERRORBARCAPSCALE=positive-number
specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

Note: This feature applies to SAS 9.4M5 and to later releases.

Default 1

Interaction This option is ignored when ERRORBARCAPSHAPE=NONE is in effect.
Tips

Use the `ERRORBARCAPSHAPE=` option to specify the cap shape.

The thickness (height) of the SERIF, LINE, and BRACKET is determined by the thickness specified in the `ERRORBARATTRS=` option. The cap line pattern is always solid.

Examples

Double the size of the error-bar caps:

```
errorbarcapscale=2
```

Reduce the size of the error-bar caps by 50%:

```
errorbarcapscale=0.5
```

`ERRORBARATTRS=style-element | style-element (line-options) | (line-options)`

specifies the attributes of the error bars that are associated with the data points.

Note: This feature applies to SAS 9.4M5 and to later releases.

Defaults

For ungrouped data, the GraphError style element contrastColor, lineStyle, and lineThickness attributes.

For grouped data, the lineThickness attribute from the GraphDataDefault style element, and the lineStyle and contrastColor attributes from the GraphData1-GraphDataN style elements.

Interaction

For this option to take effect, error bars must be displayed by the `XERRORLOWER=`, `XERRORUPPER=`, `YERRORLOWER=`, or `YERRORUPPER=` options.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Line Options” on page 1522 for available `line-options`.

`ERRORBARCAPSHAPE=SERIF | NONE`

specifies whether the error bars have a serif cap.

Note: This feature applies to SAS 9.4M5 and to later releases.

```
SERIF    NONE
```

Default

The GraphError:CapStyle style reference. If attribute CapStyle is not defined in the active style, then SERIF is the default value.

Interaction

Starting with SAS 9.4M5, when `ERRORBARCAPSCALE=` is in effect, the display width for SERIF caps is the cap's default width multiplied by the value specified in `ERRORBARCAPSCALE=`.

Tip

The appearance of the error bars is controlled by the `ERRORBARATTRS=` option.
FILLEDOUTLINEDMARKERS=TRUE | FALSE
specifies whether markers are drawn with both fill and an outline.

TRUE
draws filled markers (marker symbols with the suffix FILLED) using both fill and an outline.

When this option is TRUE, the fill color and outline color for filled markers are determined in the following ways:

- If the GROUP= option is specified, then by default, the fill color is derived from the GraphData1–GraphDataN style elements Color attribute, and the marker outlined color is derived from the GraphData1–GraphDataN style elements ContrastColor attribute.
- If the GROUP= option is not specified, then the marker fill is drawn by using the MARKERFILLATTRS= specification, and the outline is drawn by using the MARKEROUTLINEATTRS= specification.

FALSE
draws the markers using fill or an outline, but not both.

Default FALSE

Tip To specify the marker fill and outline colors for a non-grouped plot, set this option to TRUE, and then use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify the colors.

See GROUP= on page 842
MARKERFILLATTRS= on page 852
MARKEROUTLINEATTRS= on page 853
“boolean ” on page 1479 for other Boolean values that you can use.

GROUP=column | discrete-attr-var | expression
creates a separate series plot for each unique group value in the specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default Each distinct group value might be represented in the plot by a different combination of color, line pattern, and marker symbol. Lines and markers vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1–GraphDataN and GraphMissing style elements. Line thickness (for grouped and ungrouped data) is controlled by the LINEATTRS= option.

Restriction Starting with SAS 9.4M5, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to
the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBMax= to the value suggested in the note or change the line pattern to SOLID.

**Interactions**

The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of marker symbols, colors, and line patterns.

The marker size is set by the MARKERATTRS= option.

The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

When both the GROUP= and the COLORRESPONSE= options are specified, the color attributes are controlled by the COLORRESPONSE= option.

**Note**

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

**Tips**

The representations that are used to identify the groups can be overridden. For example, each distinct group value is often represented by a different line pattern, but you can use the LINEATTRS=(PATTERN=pattern) option to assign the same line pattern to all of the plot's line patterns, letting line color indicate group values. Likewise, you could use LINEATTRS=(COLOR=color) to assign the same color to all lines, letting line pattern indicate group values.

Starting with **SAS 9.4M2**, you can use the LINECOLORGROUP=, LINEPATTERNGROUP=, MARKERCOLORGROUP=, and MARKERSYMBOL= options to assign line colors, line patterns, marker colors, and marker symbols based on a different group column.

**GROUPDISPLAY=OVERLAY | CLUSTER**

specifies how marker groups are positioned for the coordinate pairs.

**OVERLAY**

draws markers for a given group value at the exact coordinate.

Depending on the data, markers at a given coordinate might overlap.

**CLUSTER**

draws markers for a given group value adjacent to each other.

**Tip**

Use the CLUSTERWIDTH= option to control the width of the clusters when CLUSTER is in effect.

**Default**

OVERLAY

**Interaction**

For this option to take effect, the GROUP= option must also be specified.
Note When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

DATA
orders the groups within a category in the group-column data order.

REVERSEDATA
orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip This option is useful when you want to reverse the category axis.

ASCENDING
orders the groups within a category in ascending order.

DESCENDING
orders the groups within a category in descending order.

Defaults
DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

Interactions
This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes
When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.
<table>
<thead>
<tr>
<th>Tips</th>
<th>Use the CLUSTERWIDTH= option to control the distance between the group markers in a cluster. Use the INDEX= option to alter the default sequence of visual attributes that is assigned to the groups.</th>
</tr>
</thead>
<tbody>
<tr>
<td>**INCLUDEMISSINGGROUP=**TRUE</td>
<td>FALSE specifies whether missing values of the group variable are included in the plot.</td>
</tr>
<tr>
<td>Default</td>
<td>TRUE</td>
</tr>
<tr>
<td>Interaction</td>
<td>For this option to take effect, the GROUP= option must also be specified.</td>
</tr>
<tr>
<td>Tip</td>
<td>The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.</td>
</tr>
<tr>
<td>See</td>
<td>“boolean ” on page 1479 for other Boolean values that you can use.</td>
</tr>
<tr>
<td>**INDEX=**positive-integer-column</td>
<td>expression specifies indices for mapping line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.</td>
</tr>
<tr>
<td>Requirements</td>
<td>The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.</td>
</tr>
<tr>
<td></td>
<td>The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.</td>
</tr>
<tr>
<td></td>
<td>All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.</td>
</tr>
<tr>
<td>Interaction</td>
<td>For this option to take effect, the GROUP= option must also be specified.</td>
</tr>
<tr>
<td>Notes</td>
<td>The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.</td>
</tr>
<tr>
<td></td>
<td>If you do not use this option, then the group values are mapped in the order of the data.</td>
</tr>
<tr>
<td>Tip</td>
<td>You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.</td>
</tr>
<tr>
<td><strong>LEGENDLABEL=</strong>&quot;string”</td>
<td>specifies a label to be used in a discrete legend for this plot.</td>
</tr>
<tr>
<td>Default</td>
<td>The Y-column label. If a label is not defined, then the Y-column name is used.</td>
</tr>
</tbody>
</table>
Restriction: This option applies only to an associated **DISCRETELEGEND** statement.

Interaction: If the **GROUP=** option is specified, then this option is ignored.

**LINEATTRS=** `style-element | style-element (line-options) | (line-options)` 

specifies the attributes of the series line.

**Defaults**

For non-grouped data, the **GraphDataDefault** style element.

For grouped data, the **ContrastColor** and **LineStyle** attributes of the **GraphData1**–**GraphDataN** style elements, and the **GraphDataDefault:LineThickness** style reference.

**Restrictions**

Starting with **SAS 9.4M5**, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option **LINEPATTERNOBSMAX=**. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase **LINEPATTERNOBSMAX=** to the value suggested in the note or change the line pattern to **SOLID**.

**Interactions**

This option’s **COLOR=** suboption overrides the default behavior for grouped data when the **LINECOLORGROUP=** option and the **COLORRESPONSE=** option are not specified. When the **COLOR=** suboption is specified without the **LINECOLORGROUP=** and **COLORRESPONSE=** options, all lines have the same color.

The **LINECOLORGROUP=** option and the **COLORRESPONSE=** option override this option’s **COLOR=** suboption.

This option’s **PATTERN=** suboption overrides the default behavior for grouped data when the **LINEPATTERNGROUP=** option is not specified. When the **PATTERN=** suboption is specified without the **LINEPATTERNGROUP=** option, all lines have the same pattern.

The **LINEPATTERNGROUP=** option overrides this option’s **PATTERN=** suboption.

**Tip**

If you specify a thickness of 2 pixels or greater for the series line, use the **LINEJOIN=** option to shape the line vertices so that the line does not extend beyond the marker symbols.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax for using a **style-element**.

“Line Options” on page 1522 for available **line-options**.

**LINECOLORGROUP=** `column | discrete-attr-var | expression` 

specifies a column that determines the line colors for a grouped plot independently of the **GROUP=** column.

**Note:** This feature applies to **SAS 9.4M2** and to later releases.

**discrete-attr-var** 

specifies a discrete attribute map variable.
The discrete attribute map variable is created by a `DISCRETEATTRVAR` statement.

**Restriction**  
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

**Note**  
A discrete attribute map variable is valid for this option starting with **SAS 9.4M3**.

When this option is specified with the `GROUP=` option, the series line colors are selected from the GraphData1–GraphDataN style elements based on the `LINECOLORGROUP=` column instead of on the `GROUP=` column.

**Default**  
The line colors are selected based on the `GROUP=` column.

**Requirement**  
The column value must be the same for all of the points that define a series line.

**Interactions**  
The `GROUP=` option must be specified for this option to have any effect.

This option overrides the `COLOR=` suboption of the `LINEATTRS=` option.

The `COLORRESPONSE=` option overrides this option.

**Note**  
If you specify a column in a SAS data set, colors are mapped to group values in data order. If you specify a column in a CAS in-memory table, colors are mapped to group values in ascending order of the group column character values or unformatted numeric values.

**Tip**  
Use the `LINEATTRS=` option to set the line thickness.

**See**  
`GROUP=` on page 842

```
LINEJOIN= MITER | ROUND | BEVEL
```

specifies how to shape the vertices of the series line.

**Note:** This feature applies to **SAS 9.4M4** and to later releases.

**MITER**  
joins the line vertices without adjustment.

**ROUND**  
rounds the apex of each line vertex, effectively smoothing the apex edges.

**BEVEL**  
bevels the apex of each line vertex, effectively trimming the apex to remove any part of the line that extends beyond the apex point.

The following figure shows the effect each value of `LINEJOIN=` on the vertex of two segments of a 10-point line. The red dot depicts the data point at the vertex.
**LINEPATTERNGROUP=**

Specifies a column that determines the line patterns for a grouped plot independently of the \texttt{GROUP=} column.

\textbf{Note:} This feature applies to SAS 9.4M2 and to later releases.

\textit{discrete-attr-var}

Specifies a discrete attribute map variable.

The discrete attribute map variable is created by a \texttt{DISCRETEATTRVAR} statement.

\textbf{Restriction} A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

\textbf{Note} A discrete attribute map variable is valid for this option starting with SAS 9.4M3.

When this option is specified with the \texttt{GROUP=} option, the series line patterns are selected from the GraphData1–GraphDataN style elements based on the \texttt{LINECOLORGROUP=} column instead of on the \texttt{GROUP=} column.

\textbf{Default} The line patterns are selected based on the \texttt{GROUP=} column.

\textbf{Requirement} The column value must be the same for all of the points that define a series line.

\textbf{Interactions} The \texttt{GROUP=} option must be specified for this option to have any effect.

This option overrides the \texttt{PATTERN=} suboption of the \texttt{LINEATTRS=} option.

\textbf{Note} If you specify a column in a SAS data set, the group-value line patterns are assigned in data order. If you specify a column in a CAS in-memory table, group-value line patterns are assigned in ascending order of the group column character values or unformatted numeric values.

\textbf{Tip} Use the \texttt{LINEATTRS=} option to set the line thickness.

\textbf{See} \texttt{GROUP=} on page 842

\texttt{LINEATTRS=}
LINETHICKNESSMAX=\textit{dimension}  
specifies the maximum line thickness when a response variable is used to
determine the line thickness. By default, this option determines the thickness of
the line that represents the maximum response column value.

\textbf{Note:} This feature applies to \textit{SAS 9.4M3} and to later releases.

\textbf{Default} \hspace{1em} Ten times the thickness that is specified by the GraphDataDefault
style element LineThickness attribute.

\textbf{Interactions} \hspace{1em} The LINETHICKNESSRESPONSE= option must be specified for
this option to have any effect.

The LINETHICKNESSMAXRESPONSE= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the LINETHICKNESSMAXRESPONSE= value are set to the value that is specified by this option.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

\textbf{Tip} \hspace{1em} Use the LINETHICKNESSMIN= option to specify the minimum line
thickness.

\textbf{See} \hspace{1em} “\textit{dimension}” on page 1480

“Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

LINETHICKNESSMAXRESPONSE=\textit{numeric} | \textit{scalar-numeric-expression}  
specifies the response value that corresponds to the maximum line thickness.

\textbf{Note:} This feature applies to \textit{SAS 9.4M3} and to later releases.

\textbf{Default} \hspace{1em} The maximum value in the response column that is specified in the LINETHICKNESSRESPONSE= option.

\textbf{Interactions} \hspace{1em} The LINETHICKNESSRESPONSE= option must be specified for
this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the LINETHICKNESSMAX= option.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

LINETHICKNESSMIN=\textit{dimension}  
specifies the minimum line thickness when a response variable is used to
determine the line thickness.
Note: This feature applies to SAS 9.4M3 and to later releases.

**Default**

The thickness specified by the GraphDataDefault style element LineThickness attribute.

**Interactions**

The LINETHICKNESSRESPONSE= option must be specified for this option to have any effect.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

**Tip**

Use the LINETHICKNESSMAX= option to specify the maximum line thickness.

**See**

“dimension” on page 1480

“Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

**LINETHICKNESSRESPONSE=**

Specifies a response column or range attribute variable that is used to map a line thickness to each group value.

Note: This feature applies to SAS 9.4M3 and to later releases.

**Default**

The GraphDataDefault style element LineThickness attribute.

**Interactions**

When the column values are all zero, all negative, or all missing, this option is ignored. In these cases, the default line thickness is used for all of the lines.

This option overrides suboption THICKNESS= in the LINEATTRS= option.

**Notes**

The line thicknesses are not reflected in a discrete legend.

The LINETHICKNESSRESPONSE= values are assumed to be constant for each group value in a grouped plot and for the entire plot in an ungrouped plot. If the LINETHICKNESSRESPONSE column has multiple values for a single GROUP value or ungrouped plot, unpredictable results might occur.

**See**

“Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

**MARKERATTRS=**

Specifies the attributes of the data markers.

**Defaults**

For non-grouped data, GraphDataDefault style element.
For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

**Interactions**

If FILLEDOUTLINEDMARKERS=TRUE, then this option's COLOR= suboption is ignored. In that case, to specify the marker fill color, use the MARKERFILLATTRS= option instead.

This option's COLOR= suboption overrides the default behavior for grouped data when the MARKERCOLORGROUP= option is not specified. When the COLOR= suboption is specified without the MARKERCOLORGROUP option, all markers have the same color, and the marker symbol alone distinguishes the markers.

The **MARKERCOLORGROUP**= option and the **COLORRESPONSE**= option override this option's COLOR= suboption.

This option's SYMBOL= suboption overrides the default behavior for grouped data when the MARKERSYMBOLGROUP= option is not specified. When the SYMBOL= suboption is specified without the MARKERSYMBOLGROUP= option, all markers have the same symbol, and the symbol color alone distinguishes the markers.

The **MARKERSYMBOLGROUP**= option overrides this option's SYMBOL= suboption.

The **TRANSPARENCY**= fill option overrides this option's **DATATRANSPARENCY**= suboption.

This option is ignored if the **DISPLAY**= option disables the display of the markers.

If the **DATASKIN**= option is in effect, then the data skin determines the marker outlines. Any outline-related settings from the current ODS style or from the marker attribute options are ignored.

**Note**

When **style-element** is specified, only the style element's MARKERSYMBOL, CONTRASTCOLOR, and MARKERSIZE attributes are used.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a **style-element**.

“Marker Options” on page 1523 for available marker-options.

**MARKERCOLORGROUP=column | discrete-attr-var | expression**

specifies a column that determines the marker colors for a grouped plot independently of the GROUP= column.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

**discrete-attr-var**

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a **DISCRETEATTRVAR** statement.
Restriction  A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Note  A discrete attribute map variable is valid for this option starting with SAS 9.4M3.

When this option is specified with the GROUP= option, the series line markers are selected from the GraphData1–GraphDataN style elements based on the MARKERCOLORGROUP= column instead of on the GROUP= column.

Default  The line marker colors are selected based on the GROUP= column.

Requirement  The column value must be the same for all of the points that define a series line.

Interactions  The DISPLAY= option must enable the display of the series line markers for this option to have any effect.

The GROUP= option must be specified for this option to have any effect.

This option overrides the COLOR= suboption of the MARKERATTRS= option.

The COLORRESPONSE= option overrides this option.

Note  If you specify a column in a SAS data set, the group-value marker colors are assigned in data order. If you specify a column in a CAS in-memory table, group-value marker colors are assigned in ascending order of the group column character values or unformatted numeric values.

Tip  Use the MARKERATTRS= option to set the marker symbol size.

See  GROUP= on page 842

MARKERATTRS= on page 850

MARKERFILLATTRS=style-element | (fill-options)

specifies the appearance of the filled markers.

Defaults  For non-grouped data, the COLOR attribute of the GraphDataDefault style element

For grouped data, the COLOR attribute of a GraphData1–GraphDataN style element

Restriction  The TRANSPARENCY= fill option is ignored. Use the MARKERATTRS= option to set the marker transparency.

Interaction  This option is in effect only when FILLEDDOUTLINEDMARKERS=TRUE and the DISPLAY= option enables fill display.

Note  When style-element is specified, only the style element’s COLOR attribute is used.
MARKEROUTLINEATTRS=\texttt{style-element | (line-options)}
specifies the appearance of the marker outlines.

<table>
<thead>
<tr>
<th>Defaults</th>
<th>For non-grouped data, the GraphOutlines style element.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For grouped data, the LineThickness attribute of the GraphOutlines style element and the ContrastColor attribute of a GraphData1–GraphDataN style element.</td>
</tr>
<tr>
<td>Restriction</td>
<td>The line style of the marker outline is always solid.</td>
</tr>
<tr>
<td>Interaction</td>
<td>This option is ignored when a data skin is applied by the current style or by the DATASKIN= option. In the latter case, the outline is set by the data skin.</td>
</tr>
<tr>
<td>Note</td>
<td>When \texttt{style-element} is specified, only the style element’s CONTRASTCOLOR and LINETHICKNESS attributes are used.</td>
</tr>
<tr>
<td>See</td>
<td>“General Syntax for Attribute Options” on page 1519</td>
</tr>
<tr>
<td></td>
<td>“Line Options” on page 1522</td>
</tr>
</tbody>
</table>

MARKERSYMBOLGROUP=\texttt{column | discrete-attr-var | expression}
specifies a column that determines the marker symbols for a grouped plot independently of the GROUP= column.

**Note:** This feature applies to SAS 9.4M2 and to later releases.

\textit{discrete-attr-var}
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

**Restriction**
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

**Note**
A discrete attribute map variable is valid for this option starting with SAS 9.4M3.

When this option is specified with the GROUP= option, the series line marker symbols are selected from the GraphData1–GraphDataN style elements based on the MARKERSYMBOLGROUP= column instead of on the GROUP= column.

**Default**
The line marker symbols are selected based on the GROUP= column.

**Requirement**
The column value must be the same for all of the points that define a series line.

**Interactions**
The GROUP= option must be specified for this option to have any effect.
This option overrides the SYMBOL= suboption of the MARKERATTRS= option.

The DISPLAY= option must enable the display of the series line markers for this option to have any effect.

**Note**
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or unformatted numeric values.

**Tip**
Use the MARKERATTRS= option to set the marker symbol size.

**See**
GROUP= on page 842
MARKERATTRS= on page 850

**NAME="string"**
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**
The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**
The string is used as the default legend label if the LEGENDLABEL= option is not used.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**
FALSE

**Restriction**
This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note**
In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

**See**
“When Plots Share Data and a Common Axis” on page 979
“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=(role-name-list)**
specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.
(role-name-list)
a space-separated list of role-name = column pairs.

Example  The following example assigns the column Obs to the user-defined role TIP:
角色名 = (TIP1=OBS)

Default  No user-defined roles

Requirement  The role names that you choose must be unique and different from the predefined roles X, Y, CURVELABEL, DATALABEL, GROUP, and INDEX.

SMOOTHCONNECT=TRUE | FALSE
specifies that a smoothed line passes through all vertices.

Default  FALSE

Interaction  Starting with **SAS 9.4M3**, this option is ignored when SPLINETYPE=QUADRATICBEZIER is in effect.

See  "boolean " on page 1479 for other Boolean values that you can use.

SPLINEPOINTS=positive-integer
specifies a multiplier to apply to the time interval that is in effect for the INTERVAL= axis option.

Note:  This feature applies to **SAS 9.4M3** and to later releases.

Default  20

Interaction  When this option is set to a non-default value, markers and data labels, when displayed, are positioned at their original data points.

SPLINETYPE=None | QUADRATICBEZIER
specifies the type of spline interpolation that is used to draw the series line.

Note:  This feature applies to **SAS 9.4M3** and to later releases.

The following figure shows an example of SPLINETYPE= values QUADRATICBEZIER and NONE.

**Figure 6.2**  **Spline Types QUADRATICBEZIER and NONE**
The SMOOTHCONNECT= option is ignored when this option is set to a value other than NONE.

Markers and data labels, when displayed, are positioned at their original data points as shown in Figure 6.5 on page 855.

**TIP=**(*role-list*) | NONE

specifies the information to display when the cursor is positioned over the series line. If this option is used, then the information specified replaces all the information that is displayed by default. You can specify roles for columns that do not contribute to the series plot along with roles that do.

(*role-list*)
an ordered, space-separated list of unique SERIESPLOT roles and user-defined roles.

SERIESPLOT roles include X, Y, CURVELABEL, DATALABEL, GROUP, and COLORRESPONSE. Define user-defined roles with the ROLENAME= option.

**Notes**

CURVELABEL is considered a role only when it is assigned a column of values. It is not considered a role and does not display data tips when it is assigned a string.

Starting with SAS 9.4M3, the COLORRESPONSE role is valid.

**Example**
The following example displays data tips for the columns assigned to the roles X and Y as well as the column Obs, which is not assigned to any pre-defined SERIESPLOT role. The Obs column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
```

**NONE**
suppresses data tips and URLs (if requested) from the plot.

**Default**
The columns assigned to these roles are automatically included in the data tip information: X, Y, DATALABEL, COLORRESPONSE, and GROUP.

**Requirement**
To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interaction**
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tip**
You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

**TIPFORMAT=**(*role-format-list*)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.
(role-format-list)
a space-separated list of role-name = format pairs.

Example
ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename ="string" pairs.

Example
ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

Default
The column label or column name of the column assigned to the role.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

URL=string-column
specifies an HTML page to display when a point or a segment of the curve is selected.

string-column
specifies a column that contains a valid HTML page reference (HREF) for each series line segment that is to have an active link.

Example
http://www.sas.com

Requirement
To generate selectable points or segments of the curve, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interactions
This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.
<table>
<thead>
<tr>
<th><strong>Note</strong></th>
<th>When you select a portion of a segment that is not an endpoint, the nearest segment endpoint is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tips</strong></td>
<td>The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding segment is selected.</td>
</tr>
<tr>
<td></td>
<td>The URL value can be the same for any X and Y pair, meaning that the same action is taken when the segment for those X and Y pairs is selected.</td>
</tr>
<tr>
<td></td>
<td>By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET=&quot;target&quot; option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.</td>
</tr>
</tbody>
</table>

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**Default** | X |

**Interaction** | The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973. |

**XERRORLOWER=numeric-column | expression**
specifies values for the lower endpoints on the X error bars. The error bars are drawn from the markers to the endpoints.

**Default** | The lower segment of the error bars is not drawn. |
| **Note** | The values are actual values, not relative values. |
| **Tips** | The appearance of the error bars is controlled by the ERRORBARATTRS= option. |

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the MARKERATTRS= option to specify a degree of transparency for the filled markers.

**XERRORUPPER=numeric-column | expression**
specifies values for the upper endpoints on the X error bars. The error bars are drawn from the markers to the endpoints.

**Default** | The upper segment of the error bars is not drawn. |
| **Note** | The values are actual values, not relative values. |
| **Tips** | The appearance of the error bars is controlled by the ERRORBARATTRS= option. |

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the
**MARKERATTRS=** option to specify a degree of transparency for the filled markers.

<table>
<thead>
<tr>
<th><strong>YAXIS=</strong></th>
<th>Y</th>
<th>Y2</th>
</tr>
</thead>
<tbody>
<tr>
<td>specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Default</strong></td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.</td>
<td></td>
</tr>
</tbody>
</table>

**YERRORLOWER=** \textit{numeric-column} \textit{expression}  
specifies values for the lower endpoints on the Y error bars. The error bars are drawn from the markers to the endpoints.

| **Default** | The lower segment of the error bars is not drawn. |
| **Note** | The values are actual values, not relative values. |
| **Tips** | The appearance of the error bars is controlled by the **ERRORBARATTRS=** option. |

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the **MARKERATTRS=** option to specify a degree of transparency for the filled markers.

**YERRORUPPER=** \textit{numeric-column} \textit{expression}  
specifies values for the upper endpoints on the Y error bars. The error bars are drawn from the markers to the endpoints.

| **Default** | The upper segment of the error bars is not drawn. |
| **Note** | The values are actual values, not relative values. |
| **Tips** | The appearance of the error bars is controlled by the **ERRORBARATTRS=** option. |

If markers are displayed in the plot, then the markers overlay the error bars. Large filled markers can obscure short error bars. To enable the error bars to show through the markers in that case, you can use the **MARKERATTRS=** option to specify a degree of transparency for the filled markers.

**Details**
The SERIESPLOT statement is typically used to show time-dependent data.

**Examples:**

**Example 1: Overlay Series Plot**
The following graph was generated by the “Example Program” on page 860:
Example Program

/* Mean temperature data for August in 2014. */
data augusttemps;
  input day ambient dewpoint apparent;
  label day="Day of Month" Ambient="Ambient"
       Apparent="Apparent" Dewpoint="Dew Point";
datalines;
1   68.74  67.96  71.14
2   66.49  66.49  68.23
3   70.05  69.41  72.70
4   73.70  70.19  76.82
5   76.09  70.89  79.89
6   78.57  70.16  81.93
7   77.10  71.02  80.98
8   74.14  70.33  77.20
9   67.16  66.66  69.01
10  70.39  68.82  72.75
11  73.75  71.53  77.40
12  74.11  72.81  78.89
13  73.36  67.32  75.66
14  70.79  61.63  71.61
15  73.40  66.76  75.61
16  76.79  70.37  80.37
17  77.64  72.23  82.17
18  76.73  74.12  81.65
19  75.44  72.92  79.85
20  78.03  73.85  83.52
21  79.32  74.57  85.02
22  78.58  75.37  84.45
23  76.80  75.15  81.79
24  73.46  67.93  75.74
25  69.71  64.37  71.18
Example 2: Grouped Series Plot

The following graph was generated by the "Example Program" on page 862:
Example Program

```
proc template;
define statgraph seriesplot;
begingraph;
  entrytitle "Tech Stock Trends";
  layout overlay /
    xaxisopts=(label="Date (mm/yy)"
      griddisplay=on gridattrs=(color=lightgray pattern=dot)
      timeopts=(tickvalueformat=mmyys5.minorticks=true
                    minortickinterval=month))
    yaxisopts=(griddisplay=on gridattrs=(color=lightgray pattern=dot)
      linearopts=(minorticks=true minortickcount=5))
  seriesplot x=date y=close / group=stock name="stocks";
  discretelegend "stocks";
endlayout;
endgraph;
run;
```

```
proc sgrender data=sashelp.stocks template=seriesplot;
  where date > "31dec1999"d;
run;
```

Example 3: Series Plot with Line-Thickness Response and Arrowheads

Starting with SAS 9.4M3, you can specify a response variable to control the thickness of the lines in your grouped series plot. You can also add arrowheads to one or both ends of your grouped or ungrouped series lines. This example creates a
Example 3: Series Plot with Line-Thickness Response and Arrowheads

grouped series plot that shows the monthly closing price for IBM, Intel, and Microsoft stock in 2001. The mean volume is computed for each stock and is used to control the thickness of the series line for each stock. The minimum line thickness is set to 2px, and the maximum line width is set to 7px. A barbed arrowhead, scaled to 1.25, is positioned at the end of each series line. In order to position the arrowheads properly, the data must be sorted in ascending order by date. The following figure shows the output.

Example Program

```sas
/* Extract the 2001 data from Sashelp.Stocks and convert Volume to millions. */
data stocks;
  set sashelp.stocks(where=(date between "1jan02"d and "31dec02"d));
  volume = volume / 1000000;
  format date MONNAME3. volume 6.2;
;
/* Compute the average volume for each stock */
proc means data=stocks noprint;
  by stock notsorted;
  var volume;
  output out=meanvolume(keep=stock meanvolume) mean=meanvolume;
run;

/* Merge the average volume data with the stock data */
data stocks;
  merge stocks meanvolume;
  by stock;
run;

/* Sort the data by Date */
```
Example 4: Series Plot with Quadratic Bézier Spline Curves

This example shows you how to create a series plot with quadratic Bézier spline curves. The SERIESPLOT statement is used to generate a grouped plot of stock closing prices in the year 2002. The SPLINETYPE=QUADRATICBEZIER option is used in the SERIESPLOT statement to specify quadratic Bézier spline curves. Here is the output for this example.
Example Program

```sas
proc template;
define statgraph seriesplot;
begingraph;
entrytitle "Stock Performance in 2002";
layout overlay /
xaxisopts=(type=discrete label="Month"
discreteopts=(tickvaluefitpolicy=thin));
seriesplot x=date y=close / name="series" group=stock
splinetype=quadraticbezier;
discretelegend "series" / across=3 displayclipped=true;
discretelegend "series" / across=3 displayclipped=true;
discretelegend "series" / across=3 displayclipped=true;
discretelegend "series" / across=3 displayclipped=true;
endlayout;
endgraph;
end;

proc sgrender data=sashelp.stocks template=seriesplot;
  where date between "1jan02"d and "31dec02"d;
  format date MONNAME3.;
run;
```

STEPPLOT Statement
Displays a series of horizontal and vertical line segments that connect observations of input data.

**Syntax**

```sas
STEPPLOT X=column | expression
Y=numeric-column | expression <options>;
```
Summary of Optional Arguments

Appearance options

ARROWHEADPOSITION= NONE | START | END | BOTH | column
specifies a position for arrowheads.

ARROWHEADSCALE= positive-number | numeric-column | expression
specifies an arrowhead scale factor based on the thickness of the arrow line.

ARROWHEADSHAPE= OPEN | FILLED | BARBED | column
specifies a shape for arrowheads.

BREAK= TRUE | FALSE
determines whether the plot line should show breaks at occurrences of missing values of the Y column.

CLUSTERWIDTH= number
on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

DATASKIN= NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the step plot lines.

DATATRANSPARENCY= number
specifies the degree of the transparency of the step lines, markers, error bars, step-line labels, and data labels, when displayed.

DISPLAY= (display-options) | STANDARD | ALL
specifies which step-line features to display.

ERRORBARATTRS= style-element | style-element (line-options) | (line-options)
specifies the attributes of the error bars that are associated with the data points.

ERRORBARCAPSCALE= positive-number
specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

ERRORBARCAPSHAPE= SERIF | NONE
specifies whether the error bars have a serif cap.

ERRORLOWER= numeric-column | expression
specifies the values of the lower endpoints on the Y error bars.

ERRORUPPER= numeric-column | expression
specifies the values of the upper endpoints on the Y error bars.

FILLEDOUTLINEDMARKERS= TRUE | FALSE
specifies whether markers are drawn with both fill and an outline.

INDEX= positive-integer-column | expression
specifies indices for mapping line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

JUSTIFY= (LEFT | CENTER | RIGHT)
specifies the location of the data point relative to the step.

LINEATTRS= style-element | style-element (line-options) | (line-options)
specifies the attributes of the step line connecting the data points.

LINETHICKNESSMAX= dimension
specifies the maximum line thickness when a response variable is used to determine the line thickness.

LINETHICKNESSMAXRESPONSE= numeric | scalar-numeric-expression
specifies the response value that corresponds to the maximum line thickness.

**LINETHICKNESSMIN=** *dimension*

specifies the minimum line thickness when a response variable is used to determine the line thickness.

**LINETHICKNESSRESPONSE=** *numeric-column | expression*

specifies a response column or range attribute variable that is used to map a line thickness to each group value.

**MARKERATTRS=** *style-element | style-element (marker-options) | (marker-options)*

specifies the attributes of the data markers.

**MARKERFILLATTRS=** *style-element | (fill-options)*

specifies the appearance of the filled markers.

**MARKEROUTLINEATTRS=** *style-element | (line-options)*

specifies the appearance of the marker outlines.

### Axes options

**CLUSTERAXIS=** *AUTO | X | Y*

specifies the axis to use for clustering groups when `GROUPDISPLAY=CLUSTER`.

**PRIMARY=** *TRUE | FALSE*

specifies that the data columns for this plot and the plot type be used for determining default axis features.

**XAXIS=** *X | X2*

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS=** *Y | Y2*

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

### Connect options

**CONNECTORDER=** *XVALUES | XAXIS*

specifies how to connect the data points to form the step line.

**JOIN=** *TRUE | FALSE*

specifies whether the steps are connected.

### Data tip options

**ROLENAME=(** *role-name-list)*

specifies user-defined roles that can be used to display information in the data tips.

**TIP=(** *role-list | NONE)*

specifies the information to display when the cursor is positioned over the step line.

**TIPFORMAT=(** *role-format-list)*

specifies display formats for tip columns.

**TIPLABEL=(** *role-label-list)*

specifies display labels for tip columns.

### Label options

**CURVELABEL=** *"string" | column | expression*

specifies a label for the step line.
CURVELABELATTRS=style-element | style-element (text-options) | (text-options)
    specifies the color and font attributes of the step-line labels.

CURVELABELLOCATION=INSIDE | OUTSIDE
    specifies the location of the step-line label relative to the plot area.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END
    specifies the position of the step-line labels relative to the step line.

CURVELABELSPLIT=TRUE | FALSE
    specifies whether to split the step-line label at the specified split characters.

CURVELABELSPLITCHAR="character-list"
    specifies one or more characters on which the step-line label can be split if needed.

CURVELABELSPLITCHARDROP=TRUE | FALSE
    specifies whether the split characters are included in the step-line label text.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
    specifies the justification of the strings that are inside the step-line label block.

DATALABEL=column | expression
    specifies a column that supplies values for the data point labels.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)
    specifies the color and font attributes of the data labels.

DATALABELPOSITION=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
    specifies the location of the data labels relative to the data points and markers, when displayed.

DATALABELSPLIT=TRUE | FALSE
    specifies whether to split the data labels at the specified split characters.

DATALABELSPLITCHAR="character-list"
    specifies one or more characters on which the data labels can be split if needed.

DATALABELSPLITCHARDROP=TRUE | FALSE
    specifies whether the split characters are included in the data labels.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
    specifies the justification of the strings that are inside the data label blocks.

LEGENDLABEL="string"
    specifies a label to be used in a discrete legend for this plot.

Midpoint options

DISCRETEOFFSET=number
    specifies an amount to offset all step lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.

GROUP=column | discrete-attr-var | expression
    creates a distinct set of lines, markers, and data labels for each unique group value of the specified column.

GROUPDISPLAY=OVERLAY | CLUSTER
    specifies whether grouped step lines and markers are overlaid or clustered around the category midpoints.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING
specifies the ordering of the groups within a category.

\texttt{INCLUDEMISSINGGROUP=TRUE | FALSE}

specifies whether missing values of the group variable are included in the plot.

**OODS options**

\texttt{URL=string-column}

specifies an HTML page to display when a step line segment is selected.

**Plot reference options**

\texttt{NAME="string"}

assigns a name to this plot statement for reference in other template statements.

**Required Arguments**

\texttt{\textit{X=}column | expression}

specifies the column of the X values.

\texttt{\textit{Y=}numeric-column | expression}

specifies the numeric column of the Y values.

**Optional Arguments**

\texttt{\textit{ARROWHEADPOSITION=} NONE | START | END | BOTH | column}

specifies a position for arrowheads.

---

\textbf{Note:} This feature applies to SAS 9.4M3 and to later releases.

\texttt{\textit{NONE}}

no arrowheads are displayed.

\texttt{\textit{START}}

an arrowhead is displayed at the starting point of each line.

\texttt{\textit{END}}

an arrowhead is displayed at the ending point of each line.

\texttt{\textit{BOTH}}

an arrowhead is displayed at each end of each line.

\texttt{\textit{column}}

specifies a column that provides an arrowhead position for each group value.

\textbf{Default} \texttt{NONE}

\textbf{Restriction} When you specify a column and the GROUP= option is in effect, the arrowhead position values are assumed to be constant for each group value. If the column has multiple values for a single group value, then only one of the arrowhead position values is used for that group.

\textbf{See} "Example 3: Series Plot with Line-Thickness Response and Arrowheads" on page 862 for an example of how to use this option.

\texttt{\textit{ARROWHEADSCALE=}positive-number | numeric-column | expression}

specifies an arrowhead scale factor based on the thickness of the arrow line.
Note: This feature applies to SAS 9.4M3 and to later releases.

**Default** 1.0

**Restriction** The arrowhead scale values are assumed to be constant for each line. If a column or expression provides multiple values for a line, only one of the values is used.

**Interaction** This option is ignored when ARROWHEADPOSITION=NONE is in effect.

**Tip** Use a factor greater than 1.0 to make a larger arrowhead.

**See** “Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

**ARROWHEADSHAPE=** OPEN | FILLED | BARBED | column

specifies a shape for arrowheads.

Note: This feature applies to SAS 9.4M3 and to later releases.

The following figure shows each of the arrowhead shapes.

![Arrowhead Shapes]

**Default** OPEN

**Restriction** When you specify a column and the GROUP= option is in effect, the arrowhead shape values are assumed to be constant for each group value. If the column has multiple values for a single group value, only one of the arrowhead shape values is used for that group.

**Interaction** This option is ignored when ARROWHEADPOSITION=NONE is in effect.

**See** “Example 3: Series Plot with Line-Thickness Response and Arrowheads” on page 862 for an example of how to use this option.

**BREAK=** TRUE | FALSE

determines whether the plot line should show breaks at occurrences of missing values of the Y column.

**Default** FALSE

**Note** When this option is set to FALSE, missing values are skipped and the line continues through the missing value and to the next point.

**See** “boolean” on page 1479 for other Boolean values that you can use.

**CLUSTERAXIS=** AUTO | X | Y

specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.
AUTO
uses the discrete axis for clustering groups when only one axis is discrete. Uses the X axis for clustering if both axes are discrete or interval.

X
Y
uses the X or Y axis for clustering groups.

Default AUTO

Interaction The GROUPDISPLAY= option must be set to CLUSTER for this option to have any effect.

CLUSTERWIDTH=number
on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

Default 0.85

Range 0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

Interactions For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

When markers are displayed for interval data and GROUPDISPLAY=CLUSTER and CLUSTERWIDTH= are in effect, the size of the markers in each cluster might be reduced to no less than 5 pixels in order to display the cluster within the smallest effective midpoint space. If you need larger markers in that case, use the MARKERATTRS= option to specify a larger marker size.

CONNECTORDER=XVALUES | XAXIS
specifies how to connect the data points to form the step line.

XVALUES
connects data points in the data order of the X column.

Note In SAS 9.4M6 and earlier releases, unexpected results might occur when XVALUES is specified and data stored in a CAS in-memory table is used. In that case, use XAXIS instead. Starting with SAS Viya 3.5 and SAS Studio 5.2, XVALUES is ignored when data stored in a CAS in-memory table is used.

XAXIS
connects data points sorted by their X values.

Tip When the input data for the step lines is not sorted by the X column, you can use XAXIS to assure the expected connect order.

Defaults XVALUES if the data is a SAS data set

XAXIS if the data is a CAS in-memory table

CURVELABEL="string" | column | expression
specifies a label for the step line.
Restrictions

When the GROUP= option is specified, "string" and expression are not valid. Use column in that case.

When the GROUP= option is not specified, column is not valid. Use "string" or expression in that case.

The line label for missing values is ignored.

Tip

The font and color attributes for the label are specified by the CURVELABELATRMS= option.

See

GROUP= on page 883

**CURVELABELATRMS=**<br>

*style-element | style-element (text-options) | (text-options)*

specifies the color and font attributes of the step-line labels. When the GROUP= option is used, each distinct group value might be represented by a different color. The series label that is associated with the group is assigned the group color. This option can be used to specify a single color for all series labels in a plot, without affecting items that have the group color, such as lines and marker symbols.

Defaults

For non-grouped data, the GraphValueText style element.

For grouped data, text color is derived from the GraphData1:ContrastColor–GraphDataN:ContrastColor style references. The font is derived from the GraphValueText style element.

Interactions

For this option to take effect, the CURVELABEL= option must also be used.

This option’s COLOR= setting overrides the colors indicated by the GROUP= option.

See

"General Syntax for Attribute Options" on page 1519 for the syntax on using a *style-element*.

"Text Options" on page 1525 for available *text-options*.

**CURVELABELLOCATION=**<br>

*INSIDE | OUTSIDE*

specifies the location of the step-line label relative to the plot area.

| INSIDE | locates the labels inside the plot area |
| OUTSIDE | locates the labels outside the plot area |

Default

INSIDE

Restrictions

OUTSIDE cannot be used when the STEPPLOT is used in multicell layouts such as LATTICE, DATAPANEL, or DATALATTICE, where axes are external to the grid.

Interactions

For this option to take effect, the CURVELABEL= option must also be specified.
This option is used in conjunction with the CURVELABELPOSITION= option to determine where the step-line labels appear. For more information, see “Location and Position of Curve Labels” on page 200.

CURVELABELPOSITION=AUTO | MAX | MIN | START | END

specifies the position of the step-line labels relative to the step line.

**AUTO**
automatically positions the step-line label near the step boundary along unused axes whenever possible (typically Y2 and X2) in order to avoid collision with tick values.

**Restriction** This option is used only when CURVELABELPOSITION=OUTSIDE.

**MAX**
forces the step-line label to appear near maximum step values (typically, upper right)

**MIN**
forces the step-line label to appear near minimum step values (typically, lower left)

**START**
forces the step-line label to appear near the beginning of the steps.

**Restriction** This option is used only when CURVELABELLOCATION=INSIDE.

**Tip**
This option is particularly useful when the step line spirals around.

**END**
forces the step-line label to appear near the end of the steps.

**Restriction** This option is used only when CURVELABELLOCATION=INSIDE.

**Tip**
This option is particularly useful when the step line spirals around.

**Defaults**
AUTO when CURVELABELLOCATION=OUTSIDE.

END when CURVELABELLOCATION=INSIDE.

**Restriction**
The AUTO setting is ignored if CURVELABELLOCATION= INSIDE is specified. The START and END settings are ignored if CURVELABELLOCATION=OUTSIDE is specified.

**Interactions**
For this option to take effect, the CURVELABEL= option must also be specified.

This option is used in conjunction with the CURVELABELLOCATION= option to determine where the step-line label appears. For more information, see “Location and Position of Curve Labels” on page 200.
When you specify TICKVALUELIST=, VIEWMAX=, or VIEWMIN= in an axis statement, the data points that are used to determine the position of the step-line label might fall outside of the graph area. In that case, the step-line label might not be displayed or might be positioned incorrectly.

Tip
When END is in effect, no attempt is made to avoid curve label collisions. If your curve labels collide, try using MAX instead.

**Note**

**CURVELABELSPLIT=TRUE | FALSE**

specifies whether to split the step-line label at the specified split characters. When a step-line label is split, the label is split on each occurrence of the specified split characters.

**Default**
FALSE. The step-line label is not split.

**Requirement**
The CURVELABEL= option must also be specified.

**Interactions**
The CURVELABELSPLITCHAR= option specifies one or more characters on which the splits occur.

This option has no effect when CURVELABELPOSITION=AUTO.

**See**
"boolean " on page 1479 for other Boolean values that you can use.

**CURVELABELSPLITCHAR=character-list**

specifies one or more characters on which the step-line label can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the step-line label. In that case, all of the specified split characters together are treated as a single split character.

When CURVELABEL= is specified and CURVELABELSPLIT=TRUE, the step-line label is split unconditionally at each occurrence of any of the specified split characters. If the step-line label does not contain any of the specified characters, then the label is not split.

"character-list"
one or more characters with no delimiter between each character and enclosed in quotation marks.

**Default**
A blank space

**Requirements**
The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no delimiters. For example, to specify the split characters a, b, and c, use the following option:

`curvelabelsplitchar="abc"`

The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interactions**
This option has no effect if CURVELABELPOSITION=AUTO.
The CURVELABELSPLITCHARDROP= option specifies whether the split characters are included in the step-line label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the CURVELABELSPLITJUSTIFY= option to specify the justification of the strings in the step-line label block.

**CURVELABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the step-line label text.

**TRUE**

drops the split characters from the step-line label text.

**FALSE**

includes the split characters in the step-line label text.

When CURVELABELSPLIT=TRUE and CURVELABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a step-line label with the following specifications:

- CURVELABELPOSITION=MAX
- CURVELABEL="Product*Group*A"
- CURVELABELSPLIT=TRUE
- CURVELABELSPLITCHARDROP=TRUE | FALSE
- CURVELABELSPLITCHAR="*"

**Note:** The horizontal line to the left of the label represents the maximum end of the step line for reference.

<table>
<thead>
<tr>
<th>CURVELABELSPLITCHARDROP=TRUE</th>
<th>Product</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CURVELABELSPLITCHARDROP=FALSE</th>
<th>Product*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group*</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

When CURVELABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When CURVELABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**

TRUE. The split characters are dropped from the step-line label.

**Requirement**

The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.
The CURVELABELSPLITCHAR= option specifies the split characters.

See "boolean " on page 1479 for other Boolean values that you can use.

CURVELABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the step-line label block.

**AUTO**

justifies the labels based on the CURVELABELPOSITION= option.

**CURVELABELPOSITION= Value | Justification**

<table>
<thead>
<tr>
<th>Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAX or END</td>
<td>LEFT</td>
</tr>
<tr>
<td>MIN or START</td>
<td>RIGHT</td>
</tr>
</tbody>
</table>

**CENTER**

**LEFT**

**RIGHT**

justifies the labels center, left, or right, as specified.

The following figure shows an example in which CURVELABELPOSITION=MAX.

Note: The horizontal line to the left of each label represents the maximum end of the step line for reference.

In this case, because CURVELABELPOSITION=MAX, AUTO left-justifies the lines of text.

**Default**

**AUTO**

**Requirement**

The CURVELABEL= option and the CURVELABELSPLIT=TRUE option must also be specified.

**Interaction**

This option has no effect if CURVELABELPOSITION=AUTO.

DATALABEL=column | expression

specifies a column that supplies values for the data point labels.

Note The label positions are adjusted to prevent the labels from overlapping.

DATALABELATTRS=style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the data labels. When the GROUP= option is used, each distinct group value might be represented by a different color. The data label that is associated with the group is assigned the group
color. This option can be used to specify a single color for all data labels in a plot, without affecting items that have the group color, such as error bars and marker symbols.

**Defaults**  
For non-grouped data, the GraphDataText style element.

For grouped data, the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

**Interactions**  
For this option to take effect, the DATALABEL= option must also be specified.

This option’s COLOR= setting overrides the colors indicated by the GROUP= option.

**Note**  
When the DATALABELPOSITION=AUTO option is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

**See**  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**DATALABELPOSITION=**  
AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT

specifies the location of the data labels relative to the data points and markers, when displayed.

**AUTO**

uses a data-label collision avoidance algorithm to position the labels.

**See**  
“LABELPLACEMENT=AUTO | GREEDY | SA” on page 33

**Note**  
When AUTO is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

**DATALABELSPLIT=**

TRUE | FALSE

specifies whether to split the data labels at the specified split characters.

**Default**  
FALSE. The data labels are not split.

**Requirement**  
The DATALABEL= option must also be specified.
The DATALABELSPLITCHAR= option specifies one or more characters on which splits can occur.

This option has no effect when DATALABELPOSITION=AUTO.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**DATALABELSPLITCHAR="character-list"**

specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

**Default**

A blank space

**Requirements**

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```
datalabelsplitchar="abc"
```

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interactions**

This option has no effect if DATALABELPOSITION=AUTO.

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels.

**TRUE**

drops the split characters from the data label.

**FALSE**

includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the
last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

The following figure shows an example of a data label with the following specifications:

- the data label text for this label is Product*Group*A
- DATALABELSPLIT=TRUE
- DATALABELSPLITCHARDROP=TRUE | FALSE
- DATALABELSPLITCHAR="***"

When DATALABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When DATALABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

**Default**

TRUE. The split characters are dropped from the data label.

**Requirement**

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interaction**

The DATALABELSPLITCHAR= option specifies the split characters.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**DATALABELSPLITJUSTIFY=** AUTO | CENTER | LEFT | RIGHT

specifies the justification of the strings that are inside the data label blocks.

**AUTO**

justifies the labels based on the DATALABELPOSITION= option.

<table>
<thead>
<tr>
<th>DATALABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

**CENTER**

LEFT

RIGHT

justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.
Note: The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.

<table>
<thead>
<tr>
<th>AUTO</th>
<th>CENTER</th>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, the bottom center of the text box is positioned at the top of the marker.

Default: AUTO

Requirement: The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interaction: This option has no effect if DATALABELPOSITION=AUTO.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the step plot lines. The following figure shows a step plot with each of the skins applied.

Default: The DATASKIN= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

Restriction: Starting with SAS 9.4M1, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

Interaction: This option overrides the BEGINGRAPH statement DATASKIN= option.
DATATRANSparency=number
specifies the degree of the transparency of the step lines, markers, error bars, step-line labels, and data labels, when displayed.

Default 0
Range 0–1, where 0 is opaque and 1 is entirely transparent

DiscreteOffset=number
specifies an amount to offset all step lines and markers from discrete X values when graphing multiple response variables side by side on a common axis.

Default 0 (no offset, all step lines and markers are centered on the discrete X values)
Range -0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. A positive offset is to the right. If the layout’s axis options set REVERSE=TRUE, then the offset direction is also reversed.
Restriction This option applies to discrete axes only. For nondiscrete axes, this option is ignored.
Tip Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

Display=(display-options) | STANDARD | ALL
specifies which step-line features to display.

(display-options)
a space-separated list of one or more options enclosed in parentheses.

Currently, only the MARKERS option is supported, which displays a step line with markers.

STANDARD displays a step line without markers.

ALL displays a step line with markers.

Default STANDARD
Tip Use the MARKERATTRS= and LINEATTRS= options to control the appearance of the line and markers.

Errorbarattrs=style-element | style-element (line-options) | (line-options)
specifies the attributes of the error bars that are associated with the data points.

Defaults For ungrouped data, the GraphError style element contrastColor, lineStyle, and lineThickness attributes.

For grouped data, the lineThickness attribute from the GraphDataDefault style element, and the lineStyle and contrastColor attributes from the GraphData1-GraphDataN style elements.
For this option to take effect, error bars must be displayed by the `ERRORLOWER=`, `ERRORUPPER=` options.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Line Options” on page 1522 for available line-options.

**ERRORBARCAPSCALE=** *positive-number*

specifies a positive number to use as a multiplier for determining the default width of the error-bar caps.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Default**

1

**Interaction**

This option is ignored when `ERRORBARCAPSHAPE=NONE` is in effect.

**Tips**

Use the `ERRORBARCAPSHAPE=` option to specify the cap shape.

The thickness (height) of the SERIF, LINE, and BRACKET is determined by the thickness specified in the `ERRORBARATTRAUS=` option. The cap line pattern is always solid.

**Examples**

Double the size of the error-bar caps:

```
errorbarcapscale=2
```

Reduce the size of the error-bar caps by 50%:

```
errorbarcapscale=0.5
```

**ERRORBARCAPSHAPE=SERIF | NONE**

specifies whether the error bars have a serif cap.

<table>
<thead>
<tr>
<th>SERIF</th>
<th>NONE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Interaction**

Starting with SAS 9.4M5, when `ERRORBARCAPSCALE=` is in effect, the display width for SERIF caps is the cap's default width multiplied by the value specified in `ERRORBARCAPSCALE=`.

**Tip**

The appearance of the error bars is controlled by the `ERRORBARATTRAUS=` option.

**ERRORLOWER=** *numeric-column | expression*

specifies the values of the lower endpoints on the Y error bars.

**Default**

The lower segment of the error bars is not drawn.

**Requirement**

The error bar values must be absolute data values, not data values relative to the value of the bar.
Tip You can use the `ERRORBARATTRS=` option to control the appearance of the error bars.

**ERRORUPPER=numeric-column | expression**
specifies the values of the upper endpoints on the Y error bars.

- **Default** The upper segment of the error bars is not drawn.
- **Requirement** The error bar values must be absolute data values, not data values relative to the value of the bar.

Tip You can use the `ERRORBARATTRS=` option to control the appearance of the error bars.

**FILLEDOUTLINEDMARKERS=TRUE | FALSE**
specifies whether markers are drawn with both fill and an outline.

- **TRUE**
  - draws filled markers (marker symbols with the suffix FILLED) using both fill and an outline.
  - When this option is TRUE, the fill color and outline color for filled markers are determined in the following ways:
    - If the GROUP= option is specified, then by default, the fill color is derived from the GraphData1–GraphDataN style elements Color attribute, and the marker outlined color is derived from the GraphData1–GraphDataN style elements ContrastColor attribute.
    - If the GROUP= option is not specified, then the marker fill is drawn by using the MARKERFILLATTRS= specification, and the outline is drawn by using the MARKEROUTLINEATTRS= specification.

- **FALSE**
  - draws the markers using fill or an outline, but not both.

- **Default** FALSE

Tip To specify the marker fill and outline colors for a non-grouped plot, set this option to TRUE, and then use the MARKERFILLATTRS= and MARKEROUTLINEATTRS= options to specify the colors.

See

- `GROUP=` on page 883
- `MARKERFILLATTRS=` on page 890
- `MARKEROUTLINEATTRS=` on page 890
- "boolean " on page 1479 for other Boolean values that you can use.

**GROUP=column | discrete-attr-var | expression**
creates a distinct set of lines, markers, and data labels for each unique group value of the specified column.

- **discrete-attr-var**
  - specifies a discrete attribute map variable.
  - The discrete attribute map variable is created by a DISCRETEATTRVAR statement.
Restriction A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default Each distinct group value might be represented in the plot by a different combination of color, line pattern, and marker symbol. Lines and markers vary according to the ContrastColor, LineStyle, and MarkerSymbol attributes of the GraphData1–GraphDataN and GraphMissing style elements. Line thickness (for grouped and ungrouped data) is controlled by the LINEATTRS= option.

Restriction Starting with SAS 9.4M5, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID.

Interactions The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of marker symbols, colors, and line patterns.

The marker size is set by the MARKERATTRS= option.

The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

Note If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

Tip The representations that are used to identify the groups can be overridden. For example, each distinct group value is often represented by a different line pattern, but the LINEATTRS=(PATTERN=pattern) option could be used to assign the same line pattern to all of the plot’s line patterns, letting line color indicate group values. Likewise, LINEATTRS=(COLOR=color) could be used to assign the same color to all lines, letting line pattern indicate group values.

See “DISCRETEATTRVAR Statement” on page 1435

GROUPDISPLAY=OVERLAY | CLUSTER
specifies whether grouped step lines and markers are overlaid or clustered around the category midpoints.

OVERLAY centers the step lines and markers for matching category values on the midpoints.

The step lines in each set of group values are superimposed on each other.

CLUSTER clusters the step lines and markers for matching category values around the midpoints on a discrete axis or around the intervals on an interval axis.
Each cluster of group values is centered at the midpoint for the category.

Default

OVERLAY

Restriction

For this option to take effect, the GROUP= option must also be specified.

Note

When you plot a SAS data set, the items for each group value are drawn in data order. When you plot a CAS in-memory table, they are drawn in ascending order of the group column character values or unformatted numeric values.

GROUPORDER=DATA | REVERSEDATA | ASCENDING | DESCENDING

specifies the ordering of the groups within a category.

DATA

orders the groups within a category in the group-column data order.

REVERSEDATA

orders the groups within a category in the reverse group-column data order.

Note: This feature applies to SAS 9.4M2 and to later releases.

Tip

This option is useful when you want to reverse the category axis.

ASCENDING

orders the groups within a category in ascending order.

DESCENDING

orders the groups within a category in descending order.

Defaults

DATA if the data is a SAS data set.

ASCENDING if the data is a CAS in-memory table. Sorting is based on nonformatted values for numeric group values or on formatted values for character group values.

Interactions

This option is ignored if the GROUP= option is not also specified.

By default, the groups in the legend are shown in the order that is specified in GROUPORDER.

Notes

When plotting a SAS data set, attributes such as color, symbol, and pattern are assigned to each group in DATA order by default. When plotting a CAS in-memory table, attributes are assigned in ascending order of the group column character values or unformatted numeric values by default.

The ASCENDING and DESCENDING settings linguistically sort the group values within each category (or X value) for display position purposes only. For numeric data, the order is based on the unformatted values. For character data, the order is based on the formatted values. The data order of the observations and the visual attributes that are assigned to the group values remain unchanged.

In SAS 9.4M6 and earlier releases, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, unpredictable results might occur. To generate consistent graphs in
that case, specify ASCENDING or DESCENDING. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA or REVERSEDATA is specified, it is ignored, and ASCENDING is used instead.

**Tip**
Use the INDEX= option to alter the default sequence of visual attributes that is assigned to the groups.

**INCLUDEMISSINGGROUP=TRUE | FALSE**
specifies whether missing values of the group variable are included in the plot.

**Default**
TRUE

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Tip**
The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

**See**
“boolean” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**
specifies indices for mapping line attributes (color, marker symbol, and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements**
The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.

All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**
For this option to take effect, the GROUP= option must also be specified.

**Notes**
The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

**Tip**
You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**JOIN=TRUE | FALSE**
specifies whether the steps are connected.
Default \textbf{TRUE}

See \textquote{boolean on page 1479} for other Boolean values that you can use.

\textbf{JUSTIFY=(LEFT | CENTER | RIGHT)}
specifies the location of the data point relative to the step.

\begin{itemize}
\item \textbf{LEFT}
\item \textbf{CENTER}
\item \textbf{RIGHT}
\end{itemize}

Default \textbf{LEFT}

\textbf{LEGENDLABEL=\textquote{string}}
specifies a label to be used in a discrete legend for this plot.

Default The Y-column label. If a label is not defined, then the Y-column name is used.

Restriction This option applies only to an associated DISCRETELEGEND statement.

Interaction If the GROUP= option is specified, then this option is ignored.

\textbf{LINEATTRS=style-element | style-element (line-options) | (line-options)}
specifies the attributes of the step line connecting the data points.

Defaults For non-grouped data, the GraphDataDefault style element.

For grouped data, the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:LineThickness style reference.

Restriction Starting with \textbf{SAS 9.4M5}, the number of vertices for a patterned line cannot exceed the maximum specified by the ODS GRAPHICS statement option LINEPATTERNOBSMAX=. The default is 10,000. If the limit is exceeded, the plot is not drawn and a note is written to the SAS log stating that the limit has been exceeded. In that case, increase LINEPATTERNOBSMAX= to the value suggested in the note or change the line pattern to SOLID.

See \textquote{General Syntax for Attribute Options on page 1519} for the syntax on using a style-element.

\textquote{Line Options on page 1522} for available line-options.

\textbf{LINETHICKNESSMAX=dimension}
specifies the maximum line thickness when a response variable is used to determine the line thickness. By default, this option determines the thickness of the line that represents the maximum response column value.

Note: This feature applies to \textbf{SAS 9.4M3} and to later releases.
Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

The LINETHICKNESSMAXRESPONSE= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the LINETHICKNESSMAXRESPONSE= value are set to the value that is specified by this option.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

Use the LINETHICKNESSMIN= option to specify the minimum line thickness.

See “dimension” on page 1480

"Example 3: Series Plot with Line-Thickness Response and Arrowheads" on page 862 for an example of how to use this option.

LINETHICKNESSMAXRESPONSE=numeric | scalar-numeric-expression

specifies the response value that corresponds to the maximum line thickness.

Note: This feature applies to SAS 9.4M3 and to later releases.

The maximum value in the response column that is specified in the LINETHICKNESSRESPONSE= option.

The LINETHICKNESSRESPONSE= option must be specified for this option to have any effect.

The thickness for all lines that exceed the maximum response value is set to the value specified in the LINETHICKNESSMAX= option.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

LINETHICKNESSMIN=dimension

specifies the minimum line thickness when a response variable is used to determine the line thickness.

Note: This feature applies to SAS 9.4M3 and to later releases.

The thickness specified by the GraphDataDefault style element LineThickness attribute.
The LINETHICKNESSRESPONSE= option must be specified for this option to have any effect.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

**Tip**
Use the LINETHICKNESSMAX= option to specify the maximum line thickness.

**See**
“dimension” on page 1480

"Example 3: Series Plot with Line-Thickness Response and Arrowheads" on page 862 for an example of how to use this option.

**LINETHICKNESSRESPONSE=**numeric-column | expression

specifies a response column or range attribute variable that is used to map a line thickness to each group value.

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**Default**
The GraphDataDefault style element LineThickness attribute.

**Interactions**
When the column values are all zero, all negative, or all missing, this option is ignored. In these cases, the default line thickness is used for all of the lines.

This option overrides suboption THICKNESS= in the LINEATTRS= option.

**Notes**
The line thicknesses are not reflected in a discrete legend.

The LINETHICKNESSRESPONSE= values are assumed to be constant for each group value in a grouped plot and for the entire plot in an ungrouped plot. If the LINETHICKNESSRESPONSE column has multiple values for a single GROUP value or ungrouped plot, unpredictable results might occur.

**See**
"Example 3: Series Plot with Line-Thickness Response and Arrowheads" on page 862 for an example of how to use this option.

**MARKERATTRS=**style-element | style-element (marker-options) | (marker-options)

specifies the attributes of the data markers.

**Defaults**
For non-grouped data, GraphDataDefault style element.

For grouped data, the MarkerSymbol and ContrastColor attributes of the GraphData1–GraphDataN style elements, and the GraphDataDefault:MarkerSize style reference.

**Interactions**
If FILLEDOUTLINEDMARKERS=TRUE, then this option’s COLOR= suboption is ignored. In that case, to specify the marker fill color, use the MARKERFILLATTRS= option instead.
This option’s COLOR= suboption overrides the default behavior for
grouped data. When the COLOR= suboption is specified in that
case, all markers have the same color, and the marker symbol
alone distinguishes the markers.

This option’s SYMBOL= suboption overrides the default behavior for
grouped data. When the SYMBOL= suboption is specified in that
case, all markers have the same symbol, and the symbol color
alone distinguishes the markers.

The TRANSPARENCY= fill option overrides this option’s
DATATRANSPARENCY= suboption.

This option is ignored if the DISPLAY= option disables the display of
the markers.

If the DATASKIN= option is in effect, then the data skin determines
the marker outlines. Any outline-related settings from the current
ODS style or from the marker attribute options are ignored.

Note When style-element is specified, only the style element’s
MARKERSYMBOL, CONTRASTCOLOR, and MARKERSIZE
attributes are used.

See “General Syntax for Attribute Options” on page 1519 for the syntax
on using a style-element.

“Marker Options” on page 1523 for available marker-options.

MARKERFILLATTRS=style-element | (fill-options)
specifies the appearance of the filled markers.

Defaults For non-grouped data, the COLOR attribute of the GraphDataDefault
style element

For grouped data, the COLOR attribute of a GraphData1–
GraphDataN style element

Restriction The TRANSPARENCY= fill option is ignored. Use the
MARKERATTRS= option to set the marker transparency.

Interaction This option is in effect only when
FILLEDOUTLINEDMARKERS=TRUE and the DISPLAY= option
enables fill display.

Note When style-element is specified, only the style element’s COLOR
attribute is used.

See “General Syntax for Attribute Options” on page 1519

“Fill Color Options” on page 1520

MARKEROUTLINEATTRS=style-element | (line-options)
specifies the appearance of the marker outlines.

Defaults For non-grouped data, the GraphOutlines style element.
For grouped data, the LineThickness attribute of the GraphOutlines style element and the ContrastColor attribute of a GraphData1–GraphDataN style element.

Restriction
The line style of the marker outline is always solid.

Interaction
This option is ignored when a data skin is applied by the current style or by the DATASKIN= option. In the latter case, the outline is set by the data skin.

Note
When style-element is specified, only the style element’s CONTRASTCOLOR and LINETHICKNESS attributes are used.

See
“General Syntax for Attribute Options” on page 1519
“Line Options” on page 1522

**NAME=“string”**
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions
The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interaction
The string is used as the default legend label if the LEGENDLABEL= option is not used.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default
FALSE

Restriction
This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note
In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See
“When Plots Share Data and a Common Axis” on page 979
“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENNAME=(role-name-list)**
specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.
(role-name-list)
a space-separated list of role-name = column pairs.

Example
The following example assigns the column Obs to the user-defined role TIP:
ROLENAME=(TIP1=OBS)

Default
No user-defined roles

Requirement
The role names that you choose must be unique and different from the predefined roles X, Y, CURVELABEL, DATALABEL, ERRORLOWER, ERRORUPPER, GROUP, and INDEX.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over the step line. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the step plot can be specified along with roles that do.

(role-list)
an ordered, space-separated list of unique STEPPLOT and user-defined roles.

STEPPLOT roles include X, Y, CURVELABEL, DATALABEL, ERRORLOWER, ERRORUPPER, and GROUP.

User-defined roles are defined with the ROLENAME= option.

Note
CURVELABEL is considered a role only when it is assigned a column of values. It is not considered a role and does not display data tips when assigned a string.

Example
The following example displays data tips for the columns assigned to the roles X and Y as well as the column Obs, which is not assigned to any pre-defined STEPPLOT role. The Obs column must first be assigned a role.

ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)

NONE
suppresses data tips and URLs (if requested) from the plot.

Default
The columns assigned to these roles are automatically included in the data tip information: X, Y, DATALABEL, ERRORLOWER, ERRORUPPER, and GROUP.

Requirement
To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip
The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.
**TIPFORMAT**=(**role-format-list**)  
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

**role-format-list**  
a space-separated list of **role-name** = **format** pairs.

Example  
```plaintext
ROLENAMES=(TIP1=Salary)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)
```

Default  
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction  
Only the roles that appear in the TIP= option are used.

Requirement  
A column must be assigned to each of the specified roles. (See the **ROLENAMES**= option.)

**TIPLABEL**=(**role-label-list**)  
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

**role-label-list**  
a space-separated list of **rolename** = **"string"** pairs.

Example  
```plaintext
ROLENAMES=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")
```

Default  
The column label or column name of the column assigned to the role.

Restriction  
Only the roles that appear in the TIP= option are used.

Requirement  
A column must be assigned to each of the specified roles. (See the **ROLENAMES**= option.)

**URL**=**string-column**  
specifies an HTML page to display when a step line segment is selected.

**string-column**  
specifies a column that contains a valid HTML page reference (HREF) for each step line segment that is to have an active link.

Example  
```plaintext
http://www.sas.com
```

Requirement  
To generate selectable segments, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interactions  
This option has no effect when **TIP**=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.
Note When selecting a portion of a segment that is not an endpoint, the nearest segment endpoint is used.

Tips The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding segment is selected.

The URL value can be the same for any X and Y pairs, meaning that the same action is taken when the segment for those X and Y pairs are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

XAXIS=X | X2 specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2 specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default Y

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details

For character columns, the X axis is always discrete. For numeric columns, both the X and Y axes are linear by default. You can change the axis type for numeric axes with the XAXISOPTS= and YAXISOPTS= options of the containing OVERLAY layout.

By default, the STEPPLOT statement uses the X values in data order. You can use the CONNECTORDER= option to change the order.

Example: STEPPLOT Statement

The following graph was generated by the “Example Program” on page 896:
**Input Data**

Here is the `DATA` step code that generates the input data for this example.

```plaintext
data Study;
input Group : $10. Time Status @@;
label Time="Time (Days)";
datalines;
Low-Risk  2569 0 Low-Risk  2506 0 Low-Risk  2409 0
Low-Risk  2218 0 Low-Risk  1857 0 Low-Risk  1829 0
Low-Risk  1562 0 Low-Risk  1470 0 Low-Risk  1363 0
Low-Risk  1030 0 Low-Risk   860 0 Low-Risk  1258 0
Low-Risk  2246 0 Low-Risk  1870 0 Low-Risk  1799 0
Low-Risk  1709 0 Low-Risk  1674 0 Low-Risk  1568 0
Low-Risk  1527 0 Low-Risk  1324 0 Low-Risk   957 0
Low-Risk   932 0 Low-Risk   847 0 Low-Risk  1535 0
Low-Risk  1447 0 Low-Risk  1384 0 Low-Risk  1514 0
Low-Risk  2204 1 Low-Risk  1063 1 Low-Risk   481 1
Low-Risk  105 1 Low-Risk   641 1 Low-Risk   390 1
Low-Risk  288 1 Low-Risk   421 1 Low-Risk    79 1
Low-Risk  748 1 Low-Risk   486 1 Low-Risk    48 1
Low-Risk  272 1 Low-Risk  1074 1 Low-Risk    38 1
Low-Risk  10 1 Low-Risk    53 1 Low-Risk    80 1
Low-Risk   35 1 Low-Risk   248 1 Low-Risk    70 1
Low-Risk  211 1 Low-Risk   219 1 Low-Risk   606 1
High-Risk 2640 0 High-Risk 2430 0 High-Risk 2252 0
High-Risk 2140 0 High-Risk 2133 0 High-Risk 1238 0
High-Risk 1631 0 High-Risk 2024 0 High-Risk 1345 0
High-Risk 1136 0 High-Risk  845 0 High-Risk  422 1
High-Risk 162 1 High-Risk   84 1 High-Risk  100 1
```

![Kaplan-Meier Survival Plot](image-url)
Example Program

Here is the program code for this example.

```sas
proc template;
  define statgraph stepplot;
  begingraph;
    entrytitle "Kaplan-Meier Survival Plot";
    layout overlay /
      yaxisopts=(linearopts=(viewmin=0 viewmax=1));
    stepplot x=Time y=Survival /
      group=Stratum name="step";
    scatterplot x=Time y=Censored / name="scat"
      legendlabel="Censored" markerattrs=(symbol=plus);
    discretelegend "step" / location=inside
      halign=right valign=top across=1;
    discretelegend "scat" /
      location=inside halign=center valign=top;
  endlayout;
  endgraph;
end;
run;

ods graphics;
ods exclude all;
ods output survivalplot=plotdata;
proc lifetest data=Study plots=(survival);
  time Time * Status(0);
  survival;
  strata Group;
run;
ods select all;

proc sgrender data=plotdata template=stepplot;
run;
```
SURFACEPLOTPARM Statement

Creates a three-dimensional surface representing a response variable evaluated over a grid of X and Y values.

Restriction: The SURFACEPLOTPARM statement does not support data tips.

Requirements: For surface plots, the input data should form an evenly spaced grid of horizontal values (X and Y) and one or more vertical values (Z) for each combination.

If the plot data is a SAS data set, the input data must be sorted by Y and X in order to obtain the correct lighting. If the plot data is a CAS in-memory table, the data is sorted automatically before it is plotted.

Syntax

SURFACEPLOTPARM X=numeric-column | expression
Y=numeric-column | expression
Z=numeric-column | expression <options>;

Summary of Optional Arguments

Appearance options

COLORMODEL=style-element | (color-list)
specifies a color ramp that is to be used with the COLORRESPONSE= or SURFACECOLORGRADIENT= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
starting with SAS 9.4M2, specifies the column or range attribute map variable to use to determine the surface colors.

DATATRANSPARENCY=number
specifies the degree of the transparency of the surface.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the color of the filled surface or the wire-frame mesh.

REVERSECOLORMODEL=TRUE | FALSE
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

SURFACECOLORGRADIENT=numeric-column | range-attr-var
specifies the column or range attribute map variable that is used to determine the surface colors (in SAS 9.4M1 and earlier releases).

SURFACETYPE=FILLGRID | FILL | WIREFRAME
specifies how the surface is displayed.

Axes options

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features.

Label options

LEGENDLABEL="string"
specifies a label to be used in a discrete legend for this plot.
Plot reference options

NAME="string"
assigns a name to this plot statement for reference in other template statements.

Required Arguments

X=numeric-column | expression
specifies the X coordinates of the grid.

Y=numeric-column | expression
specifies the Y coordinates of the grid.

Z=numeric-column | expression
specifies the height of response values.

Note The input data should be sorted by both 1) the Y column and 2) the X column. The sort direction for Y should be ascending. The sort direction of X be either ascending or descending.

Optional Arguments

COLORMODEL=style-element | (color-list)
specifies a color ramp that is to be used with the COLORRESPONSE= or SURFACECOLORGRADIENT= option.

style-element
specifies the name of a style element.

The style element should contain these style attributes:

STARTCOLOR specifies a color for the smallest data value of the COLORRESPONSE= or SURFACECOLORGRADIENT= column.

NEUTRALCOLOR specifies a color for the midpoint of the range of the COLORRESPONSE= or SURFACECOLORGRADIENT= column.

ENDCOLOR specifies a color for the highest data value of the COLORRESPONSE= or SURFACECOLORGRADIENT= column.

(color-list)
specifies a space-separated list of colors to use in the color ramp.

You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color.

The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

See “color ” on page 1480

Default The ThreeColorRamp style element.

Interaction For this option to have any effect, the COLORRESPONSE= or SURFACECOLORGRADIENT= option must also be used.
Tip: The `REVERSECOLORMODEL=` option can be used to reverse the start and end colors of the ramp assigned to the color model.

**COLORRESPONSE=** `numeric-column | range-attr-var | expression`

starting with **SAS 9.4M2**, specifies the column or range attribute map variable to use to determine the surface colors.

*Note:* Starting with **SAS 9.4M2**, the `COLORRESPONSE=` option replaces the `SURFACECOLORGRADIENT=` option. The syntax and functionality are the same. The `SURFACECOLORGRADIENT=` option is still honored, but the `COLORRESPONSE=` option is preferred.

`range-attr-var`

specifies a range attribute map variable that is defined in a `RANGEATTRVAR` statement.

"**RANGEATTRVAR Statement**"

**Restriction**

A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

**See**

"**RANGEATTRVAR Statement**" on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the `COLORMODEL=` option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

You can use this option to add a second response variable to an analysis. For an example, see "Example 1: Surface Plot" on page 901.

**Interaction**

Suboption `COLOR=` in the `FILLATTRS=` option overrides this option for the fill colors.

**Tip**

To display a legend with this option in effect, use a `CONTINUOUSLEGEND` statement.

**DATATRANSPARENCY=** `number`

specifies the degree of the transparency of the surface.

**Default**

0

**Range**

0–1, where 0 is opaque and 1 is entirely transparent

**FILLATTRS=** `style-element | style-element (fill-options) | (fill-options)`

specifies the color of the filled surface or the wire-frame mesh.

**Default**

The `GraphDataDefault:Color` style reference.

**Interaction**

The `COLORRESPONSE=` or `SURFACECOLORGRADIENT=` option is ignored if this option is specified.

**See**

"**General Syntax for Attribute Options**" on page 1519 for the syntax on using a `style-element`.

"**Fill Color Options**" on page 1520 for available `fill-options`. 
**LEGENDLABEL=“string”**
specifies a label to be used in a discrete legend for this plot.

**Default**  
The Z-column label. If a label is not defined, then the Z-column name is used.

**Restriction**  
This option applies only to an associated DISCRETELEGEND statement.

**NAME=“string”**
assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**  
The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**  
The string is used as the default legend label if the LEGENDLABEL= option is not used.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an OVERLAY3D layout contribute to a common axis.

**Default**  
FALSE

**See**  
"When Plots Share Data and a Common Axis" on page 979

"boolean “ on page 1479 for other Boolean values that you can use.

**REVERSECOLORMODEL=TRUE | FALSE**
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

**Default**  
FALSE

**See**  
COLORMODEL=

"boolean “ on page 1479 for other Boolean values that you can use.

**SURFACECOLORGRADIENT=numeric-column | range-attr-var**
specifies the column or range attribute map variable that is used to determine the surface colors (in SAS 9.4M1 and earlier releases).

**Note:** Starting with SAS 9.4M2, the SURFACECOLORGRADIENT= option is deprecated and replaced with the COLORRESPONSE= option. The syntax and functionality are the same. The SURFACECOLORGRADIENT= option is still honored, but the COLORRESPONSE= option is preferred.

**See**  
COLORRESPONSE= on page 899

**SURFACETYPE=FLGGRID | FLILL | WIREFRAME**
specifies how the surface is displayed.
FILLGRID
    specifies a filled surface with superimposed grid lines
FILL
    specifies a filled surface without grid lines
WIREFRAME
    specifies an unfilled surface with grid lines
Default:  FILLGRID

Details

The SURFACEPLOTPARM statement assumes that the Z response values have been provided for a uniform X-Y grid. Missing Z values leave a “hole” in the surface. The observations in the input data set should form an evenly spaced grid of horizontal (X and Y) values and one vertical (Z) value for each of these combinations. The observations should be in sorted order of Y and X to obtain an accurate plot.

The G3GRID procedure (requires a SAS/GRAPH license) can be used to interpolate the necessary values to produce a data set with nonmissing Z values for every combination of X and Y. The G3GRID procedure can also smooth data with spline interpolations. For further details, see the documentation for PROC G3GRID in the SAS/GRAPH: Reference.

The KDE procedure can produce an output data set of gridded X-Y values where the Z value is computed to be a Kernel Density Estimate of the distribution of X and Y. For further details, see the documentation for PROC KDE in the SAS/STAT user’s guide.

Examples:

Example 1: Surface Plot

The following graph was generated by the “Example Program” on page 902:
Example Program

Here is the code for this example. The COLORRESPONSE= option is valid starting with SAS 9.4M2. For prior releases, use the SURFACECOLORGRADIENT= option instead.

```sas
proc template;
   define statgraph surfaceplotparm;
      begingraph;
         entrytitle "Surface Plot of Lake Bed";
         layout overlay3d / cube=false;
         surfaceplotparm x=length y=width z=depth /
            reversecolormodel=true
            colorresponse=depth
            colormodel=twocoloraltramp;
      endlayout;
      endgraph;
   end;

/* create gridded data for surface
 * proc g3grid is a sas/graph procedure */
proc g3grid data=sashelp.lake out=gridded;
   grid width*length = depth / naxis1=75 naxis2=75;
run;

proc sgrender data= gridded template=surfaceplotparm;
run;
```

Example 2: Spline Interpolation

The following graph was generated by the “Example Program” on page 903:
Input Data

Here is the SAS code that generates the data for this example.

```sas
data nums;
  do i=1 to 30;
    X=10*ranuni(33)-5;
    Y=10*ranuni(33)-5;
    Z=sin(sqrt(x*x+y*y));
    output;
  end;
run;
proc g3grid data=nums out=gridded;
  grid y*x=z / spline
    axis1=-5 to 5 by 0.1
    axis2=-5 to 5 by 0.1;
run;
proc sort data=gridded; by y x; run;
```

Example Program

```sas
proc template;
  define statgraph g3grid_surface;
  begingraph;
    entrytitle "Spline Interpolation";
    layout overlay3d;
      surfaceplotparm x=x y=y z=z /
        surfacetypetype=fill;
  endlayout;
```
Example 3: Kernel Density Estimate

The following graph was generated by the "Example Program" on page 905:

Input Data

Here is the SAS code that generates the data for this example.

```sas
data nums;
  do i=1 to 30;
    X=10*ranuni(33)-5;
    Y=10*ranuni(33)-5;
    Z=sin(sqrt(x*x+y*y));
    output;
  end;
run;
proc kde data=nums;
  bivar x y / ngrid=100
    gridl=-5 gridu=5
    out=binned(rename=(value1=X value2=Y));
run;
proc sort data=binned; by y x;
  label x="X" y="Y";
run;
```
Example Program

```sas
proc template;
  define statgraph kde_surface;
  begingraph;
    entrytitle "Kernel Density Estimate";
    layout overlay3d;
      surfaceplotparm x=x y=y z=density / surfacetype=fill;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=binned template=kde_surface;
run;
```

TEXTPLOT Statement

Displays text values at specific X and Y locations in the graph.

**Note:** This feature applies to [SAS 9.4M2](https://www.sas.com) and to later releases.

**Tip:** Use the TEXTPLOT statement, rather than the SCATTERPLOT statement with the MARKERCHARACTER= option, when you want more control over the appearance of the text. The TEXTPLOT statement enables you to rotate the text to any angle, manage the text position, split the text into multiple lines, display a bounding box around the text, add a back-light effect to the text, and so on.

**Syntax**

TEXTPLOT  
X=column | expression  
Y=column | expression  
TEXT=column | expression  
=options>

**Summary of Optional Arguments**

**Appearance options**

- **BACKLIGHT=**number | AUTO  
  Specifies a back-light effect for the marker text.

- **CLUSTERWIDTH=**number  
  On a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

- **COLORMODEL=**color-ramp-style-element | (color-list)  
  Specifies a color ramp to use with the COLORRESPONSE= option.

- **COLORRESPONSE=**numeric-column | range-attr-var | expression  
  Specifies the column or range attribute map variable to use to determine the text-marker text colors.

- **CONTRIBUTEOFFSETS=**ALL | NONE | (axis-offset-list)

---
specifies whether space requirements for this plot contribute to the calculation of the axis offsets.

**DATATRANSPERANCY=number**
specifies the degree of the transparency of the text.

**DISPLAY=(display-options) Standard | All**
specifies the text features to display.

**FILLATTRS=style-element | (fill-options)**
specifies the appearance of the filled areas of the text.

**INDEX=positive-integer-column | expression**
specifies indices for mapping marker text color to one of the GraphData1–GraphDataN style elements.

**OUTLINEATTRS=style-element | style-element(line-options) | (line-options)**
specifies the appearance of the text-marker outlines.

**PAD=dimension | (pad-options)**
specifies the amount of extra space to add inside the text-marker border.

**REVERSECOLORMODEL=TRUE | FALSE**
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

**TEXTATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font properties of the text-marker text.

### Axes options

**CLUSTERAXIS=AUTO | X | Y**
specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features.

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

### Data tip options

**ROLENAMESPACE=(role-name-list)**
specifies user-defined roles that can be used to display information in the data tips.

**TIP=(role-list) | NONE**
specifies the information to display when the cursor is positioned over the text values.

**TIPFORMAT=(role-format-list)**
specifies display formats for tip columns.

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns.

### Experimental options

**OUTFILE=fileref | "filename"**
specifies a file for storing information about the text bounding-box for each text value in the column specified in the OUTID= option.

**OUTID=column | expression**
specifies a column that contains text values to write to the file specified in the OUTFILE= option.

**Label options**

LEGENDLABEL="string"

specifies a label to be used in a discrete legend for this plot.

**Midpoint options**

DISCRETEOFFSET=number

specifies the distance to offset all text values from discrete X values, discrete Y values, or both.

GROUP=column | discrete-attrib-var | expression

creates a separate text value for each unique group value in the specified column.

GROUPDISPLAY=OVERLAY | CLUSTER

specifies how marker groups are positioned for the coordinate pairs.

INCLUDEMISSINGGROUP=TRUE | FALSE

specifies whether missing values of the group variable are included in the plot.

**ODS options**

URL=string-column

specifies an HTML page to display when a text value is selected.

**Plot reference options**

NAME="string"

assigns a name to this plot statement for reference in other template statements.

**Text options**

FORMAT=format

specifies a SAS format or a user-defined format for the TEXT= column.

POSITION= BOTTOM | BOTTOMLEFT | BOTTOMRIGHT | CENTER | LEFT | RIGHT | TOP | TOPLEFT | TOPRIGHT | keyword-column

specifies the position of the text value with respect to the location of the data point.

POSITIONOFFSETX=number | numeric-column

specifies the positive or negative X offset for an individual text item.

POSITIONOFFSETY=number | numeric-column

specifies the positive or negative Y offset for an individual text item.

ROTATE=number | numeric-column

specifies the angle of rotation, in degrees, for the text values.

SIZEMAX=dimension

specifies the maximum font size for a text marker when a response variable is used to size the text-marker font.

SIZEMAXRESPONSE=numeric | scalar-numeric-expression

specifies the response value that corresponds to the maximum font size for text markers.

SIZEMIN=dimension

specifies the minimum font size for text markers when a response variable is used to size the font for text values.

SIZERESPONSE=numeric-column | numeric-expression
specifies a response column that is used to determine the font size for each text value.

SPLITCHAR="character-list"
specifies one or more characters on which the text-marker text can be split.

SPLITDROP=TRUE | FALSE
specifies whether the split characters are included in the text-marker text.

SPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the lines of text in the text-marker text blocks.

SPLITPOLICY=NONE | SPLIT | SPLITALWAYS
specifies a policy for avoiding collisions among the text values.

SPLITWIDTH=AUTO | width-in-characters
specifies the maximum width of each split line.

STRIP=TRUE | FALSE
specifies whether leading and trailing blanks should be stripped from the text-marker text before it is displayed.

VCENTER=BBOX | BASELINE
specifies whether the text is vertically centered with respect to the text bounding box or to the text baseline.

Required Arguments

X=column
specifies the column for the X values.

Y=column
specifies the column for the Y values.

TEXT=column
specifies the column for the text values that are to be used for the markers.

Optional Arguments

BACKLIGHT=number | AUTO
specifies a back-light effect for the marker text. The effect is applied only to the marker text.

number
specifies the degree of the back-light effect.

Range 0–1, where 0 specifies no effect and 1 specifies maximum effect

AUTO
the system selects an appropriate level for the back-light effect.

If the GROUP= or COLORRESPONSE= option is in effect, BACKLIGHT=0.75. Otherwise, BACKLIGHT=0.5.

The following figure shows the effect on the text of an outlined, filled text marker.
The back light is based on text color. For dark colors, a contrasting white back-light effect is used. For lighter colors, a contrasting black back-light effect is used. The following figure shows the two back-light types when BACKLIGHT=1.

<table>
<thead>
<tr>
<th>Black Text</th>
<th>Light-Gray Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Black Text Image" /></td>
<td><img src="image2.png" alt="Light-Gray Text Image" /></td>
</tr>
</tbody>
</table>

**Default**

0 (no back-light effect)

**Restriction**

Vector graphics output cannot be generated when the back-light effect is applied. If you request vector graphics output and enable the back-light effect, an image is generated instead.

**Note**

If the background color is white, the white backlight effect for dark text colors is not visible. Conversely, if the background color is black, the black backlight effect for light text colors is not visible.

**Tip**

The BACKLIGHT= option is most effective when the text color has a low level of contrast with the background or when the background is cluttered.

**CLUSTERAXIS=**

<table>
<thead>
<tr>
<th>AUTO</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
</table>

specifies the axis to use for clustering groups when GROUPDISPLAY=CLUSTER.

**AUTO**

uses the discrete axis for clustering groups when only one axis is discrete.

Uses the X axis for clustering if both axes are discrete or interval.

**X**

**Y**

uses the X or Y axis for clustering groups.

**Default**

AUTO

**Interaction**

The GROUPDISPLAY= option must be set to CLUSTER for this option to have any effect.

**CLUSTERWIDTH=**

`number`

on a discrete axis, specifies the width of the group clusters as a fraction of the midpoint spacing. On an interval axis, specifies the width of the group clusters as a fraction of the minimum interval between adjacent data values.

**Default**

0.85

**Range**

0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

**Interaction**

For this option to take effect, the GROUP= option must also be specified, and the GROUPDISPLAY= option must be set to CLUSTER.

**COLORMODEL=**

`color-ramp-style-element | (color-list)`

specifies a color ramp to use with the COLORRESPONSE= option.
The style element should contain these style attributes:

**STARTCOLOR**
- specifies the color for the smallest data value of the COLORRESPONSE= column.

**NEUTRALCOLOR**
- specifies the color for the midpoint of the range of the COLORRESPONSE= column.

**ENDCOLOR**
- specifies the color for the highest data value of the COLORRESPONSE= column.

**(color-list)**
- specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

**Requirement** The list of colors must be enclosed in parentheses.

**See** “color” on page 1480

**Default** The ThreeColorAltRamp style element.

**Interaction** For this option to take effect, the COLORRESPONSE= option must also be specified.

**Tip** To reverse the start and end colors of the ramp that is assigned to the color model, use the REVERSECOLORMODEL= option.

**COLORRESPONSE=**
- numeric-column | range-attr-var | expression
- specifies the column or range attribute map variable to use to determine the text-marker text colors.

**range-attr-var**
- specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

**“RANGEATTRVAR Statement”**

**Restriction** A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

**Tip** For each range in the attribute map, the RANGECOLOR= or RANGECOLORMODEL= option in the RANGE statement determines the bounding-box fill colors. The RANGEALTCOLOR= or RANGEALTCOLORMODEL= option determines the text colors and the bounding-box outline colors.

**See** “RANGEATTRVAR Statement” on page 1480

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLOORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.
You can use this option to add a second response variable to an analysis.

**Interactions**

When the `GROUP=` option is specified with the `COLORRESPONSE=` option, the `GROUP=` option is ignored.

Prior to **SAS 9.4M3**, suboption `COLOR=` in the `TEXTATTRS=`, `FILLATTRS=`, and `OUTLINEATTRS=` options overrides the color attributes that are normally determined by this option.

Starting with **SAS 9.4M3**, interactions between this option and suboption `COLOR=` in the `TEXTATTRS=`, `FILLATTRS=`, and `OUTLINEATTRS=` options depend on the `DISPLAY=` option settings. See "Response Colors in a Text Plot" on page 929.

**Note**

The gradient in a continuous legend for this plot reflects the text colors.

**Tips**

To display a legend with this option in effect, use a `CONTINUOUSLEGEND` statement.

Use the `OUTLINEATTRS=` option to set the text bounding-box color to a fixed color.

Use the `FILLATTRS=` option to set the text bounding-box fill color to a fixed color or to modify the fill transparency.

Prior to **SAS 9.4M3**, when fill is displayed and the `COLORRESPONSE=` option is in effect, the text color and the fill color are derived from the color gradient, which makes the text unreadable. In that case, use the `BACKLIGHT=` option to add a backlight effect to the text, or use the `TEXTATTRS=` or `FILLATTRS=` option to specify a different text or fill color.

**CONTRIBUTEOFFSETS=ALL | NONE | (axis-offset-list)**

specifies whether space requirements for this plot contribute to the calculation of the axis offsets. This plot's layout container queries each of its plots for a preferred offset and includes all of the offsets in the axis offset calculations. If the `DATALABEL=` or `MARKERCHARACTER=` option is specified for this plot, this plot might request a preferred offset that prevents the clipping of any data labels or marker character strings that appear at the ends of the axes. The requested offset is based on the longest string. If the label or marker character lengths vary significantly, the result is wasted space when the shorter strings appear near the ends of the axes. In that case, you can use the `CONTRIBUTEOFFSETS=` option to modify or eliminate this plot's contribution to the offset calculations in order to reclaim that space.

**ALL**

the space requirements for this plot are contributed to the axis offset calculations.

**NONE**

the space requirements for this plot are not contributed to the axis offset calculations.

**(axis-offset-list)**

a space-delimited list of specific contributions that this plot makes to the axis offset calculations.

The list is one or more of the following values, enclosed in parentheses:
XMAX
  the space requirements for this plot are contributed to the X-axis offset
calculation for the maximum end.

XMIN
  the space requirements for this plot are contributed to the X-axis offset
calculation for the minimum end.

YMAX
  the space requirements for this plot are contributed to the Y-axis offset
calculation for the maximum end.

YMIN
  the space requirements for this plot are contributed to the Y-axis offset
calculation for the minimum end.

Default    ALL

Interaction Offsets that are set in the layout axis options are always honored,
regardless of the setting on this option.

Note       This option does not affect offset requests from other plots.

DATATRANSPARENCY=number
  specifies the degree of the transparency of the text.

  Default    0
  Range      0–1, where 0 is opaque and 1 is entirely transparent

DISCRETEOFFSET=number
  specifies the distance to offset all text values from discrete X values, discrete Y
values, or both.

  Default    0 (no offset, all text values are centered on the discrete X values, or
discrete Y values, or both)
  Range      –0.5 to +0.5, where 0.5 represents half the distance between
discrete tick marks. A positive offset is to the right on discrete X
values and is up on discrete Y values. If the layout’s axis options set
REVERSE=TRUE, then the offset direction is also reversed.
  Restriction This option applies to discrete axes only. For nondiscrete axes, this
option is ignored.
  Tip        Setting the discrete offset for the plots does not affect the axis
minimum and maximum offsets. In some cases, setting a discrete
offset can cause clipping at each end of the axis. In those cases, use
the OFFSETMIN= and OFFSETMAX= axis options to increase the
axis minimum and maximum offsets to accommodate the discrete
offset.

DISPLAY=(display-options) | STANDARD | ALL
  specifies the text features to display.

(display-options)
  a space-separated list of one or more display options, enclosed in
parentheses.
  The list can include one or more of the following options:
FILL
  displays filled bounding boxes around the text

Tip  Use the FILLATTRS= option to modify the appearance of the bounding-box fill.

OUTLINE
  displays outlined bounding boxes around the text

Tip  Use the OUTLINEATTRS= option to modify the appearance of the bounding-box outline.

STANDARD
  displays the text only

ALL
  specifies FILL and OUTLINE

Default  STANDARD

Restriction  Vector graphics output cannot be generated when FILL or OUTLINE is displayed. If you request vector graphics output and specify DISPLAY=FILL or DISPLAY=OUTLINE, an image is generated instead.

Tip  When fill is displayed and the COLORRESPONSE= option is in effect, a low contrast between the fill color and the text color can make some of the text difficult to read or unreadable. In that case, use the TRANSPARENCY= suboption to adjust the fill transparency or use the BACKLIGHT= option to add a backlight effect to the text.

FILLATTRS=style-element | (fill-options)
  specifies the appearance of the filled areas of the text. When fill options are specified, only the COLOR= and TRANSPARENCY= suboptions are honored.

Defaults  For non-grouped data, the Color attribute of the GraphDataDefault style element.

For grouped data, the Color attribute of the GraphData1–GraphDataN style elements.

Interactions  For this option to have any effect, the fill must be enabled by the ODS style or by the DISPLAY= option.

When this option’s COLOR= suboption is specified with the GROUP= or COLORRESPONSE= option, the bounding-box fill color is set to the COLOR= specification for all of the text values.

Tip  The DATATRANSPARENCY= option sets the transparency for the text-marker text, fill, and outlines. You can combine this option with DATATRANSPARENCY= to set one transparency for the text and outlines but a different transparency for the fills. Example:

  datatransparency=0.2 fillattrs=(transparency=0.6)

See  “General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element.

“Fill Color Options” on page 1520 for available fill-options.
FORMAT=\textit{format}
specifies a SAS format or a user-defined format for the TEXT= column.

\textbf{Default} The format that is in effect for the column that is specified in the TEXT= argument. If no format is in effect, BEST6 is used for numeric columns.

\textbf{Note} Not all of the SAS formats are supported. See Appendix 5, “SAS Formats Not Supported,” on page 1541.

\textbf{GROUP=}\textit{column} | \textit{discrete-attr-var} | \textit{expression}
creates a separate text value for each unique group value in the specified column.

\textit{discrete-attr-var}
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a \texttt{DISCRETEATTRVAR} statement.

\textbf{Restriction} A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

\textbf{Tip} For each value in the discrete attribute map, the VALUE statement TEXTATTRS= option determines the text color, the LINEATTRS= option determines the bounding-box outline color, and the FILLATTRS= option determines the bounding-box fill color and transparency.

\textbf{Interactions} The group values are mapped in the order in which they appear in the data.

The \texttt{INCLUDEMISSINGGROUP} option controls whether missing group values are considered a distinct group value.

The \texttt{COLORRESPONSE=} option overrides the group settings for the text color of the text value. In that case, text color is set according to the gradient.

The \texttt{COLOR=} suboption of the \texttt{TEXTATTRS=} option overrides the group settings for the text color of the text value. In that case, the text color for all of the text values is set to the \texttt{COLOR=} specification.

\textbf{Notes} The legend entries for this plot reflect the text colors.

By default, for each text value, the bounding-box outline color is set to the text color.

If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

\textbf{Tip} Use the \texttt{OUTLINEATTRS=} option to set the text bounding-box color to a fixed color.
GROUPDISPLAY=OVERLAY | CLUSTER
specifies how marker groups are positioned for the coordinate pairs.

OVERLAY
draws text values for a given group value at the exact coordinate.
Depending on the data, markers at a given coordinate might overlap.

CLUSTER
draws text values for a given group value adjacent to each other.

Restriction CLUSTER is supported only when at least one axis is discrete.

Default OVERLAY

Note When you plot a SAS data set, the items for each group value are
drawn in data order. When you plot a CAS in-memory table, they are
drawn in ascending order of the group column character values or
unformatted numeric values.

Tip Use the CLUSTERWIDTH= option to control the width of the clusters
when CLUSTER is in effect.

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Default TRUE

Interaction For this option to take effect, the GROUP= option must also be
specified.

Tip The attributes of the missing group value are determined by the
GraphMissing style element unless a discrete attribute map is in
effect, the INDEX= option is used, the MISSING= system option
changes the default missing character, or a user-defined format is
applied to the group value. In those cases, the attributes of the
missing group value are determined by a GraphData1–GraphDataN
style element instead of by the GraphMissing style element.

See “boolean ” on page 1479 for other Boolean values that you can use.

INDEX=positive-integer-column | expression
specifies indices for mapping marker text color to one of the GraphData1–
GraphDataN style elements.

Requirements The column or expression value must be an integer value of 1 or
greater. Otherwise, this option is ignored.

The positive-integer column must not contain missing values.
Otherwise, the entire column is invalidated and this option is
ignored.

All of the indexes for a specific group value must be the same.
Otherwise, the results are unpredictable.

Interaction For this option to take effect, the GROUP= option must also be
specified.
Notes

The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.

If you do not use this option, then the group values are mapped in the order of the data.

Tip

You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL=“string”**

specifies a label to be used in a discrete legend for this plot.

Default

The TEXT-column label. If a label is not defined, then the TEXT-column name is used.

Restriction

This option applies only to an associated DISCRETELEGEND statement.

Interaction

If the GROUP= option is specified, then this option is ignored.

**NAME=“string”**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

Restrictions

The string is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**OUTFILE=fileref | “filename”**

specifies a file for storing information about the text bounding-box for each text value in the column specified in the OUTID= option. The information is written in the comma-separated value (CSV) format.

**CAUTION**

OUTFILE= is an experimental option that is available in SAS 9.4M3. Do not use this option in production jobs.

Interaction

The OUTID= option must be specified for this option to have any effect.

Note

This option is a specialized feature for users who want to customize the placement of the text in a text plot.

Tip

Use the IMPORT procedure to import the CSV values into a SAS data set.

See

“Customizing Text Marker Placement (Experimental)” on page 930
OUTID=column | expression
specifies a column that contains text values to write to the file specified in the OUTFILE= option.

**CAUTION**
OUTID= is an experimental option that is available in **SAS 9.4M3**. Do not use this option in production jobs.

**Interaction**
The OUTFILE= option must be specified for this option to have any effect.

**Note**
This option is a specialized feature for users who want to customize the placement of the text in a text plot.

**See**
“Customizing Text Marker Placement (Experimental)” on page 930

OUTLINEATTRS=style-element | style-element(line-options) | (line-options)
specifies the appearance of the text-marker outlines.

**Defaults**
For non-grouped data, the ContrastColor attribute of the GraphOutlines style element.

For grouped data, text values use the ContrastColor attribute of the GraphData1–GraphDataN style elements. If the COLORRESPONSE= option is specified, the outline colors vary according to the color gradient.

**Restriction**
This option uses only the color specification in the style element or line options. The line pattern and line thickness specifications are ignored.

**Interactions**
For this option to have any effect, outlines must be enabled by the ODS style or by the DISPLAY= option.

When the COLOR= suboption is specified with the GROUP= option or with the COLORRESPONSE= option, for all of the text values, the bounding-box border color is set to the COLOR= specification.

**Note**
This plot’s legend entries reflect the marker text color.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element.

COLOR= on page 1522

PAD=dimension | (pad-options)
specifies the amount of extra space to add inside the text-marker border.

dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the text-marker border.

(pad-options)
a space-separated list of one or more pad options, enclosed in parentheses.
The list can include one or more of the following name-value-pair options:

LEFT=dimension TOP=dimension
RIGHT=\textit{dimension} \quad \text{BOTTOM}=\textit{dimension}

**Default**

Padding is a fraction of the font height.

**Note**

Sides that are not assigned padding are padded with the default amount.

**Tips**

This option is meaningful only when the DISPLAY= option displays fills, outlines, or both.

Use \textit{pad-options} to create non-uniform padding.

**Note**

The default units for \textit{dimension} are pixels.

**See**

"\textit{dimension}" on page 1480

\textbf{POSITION=\textit{BOTTOM} | \textit{BOTTOMLEFT} | \textit{BOTTOMRIGHT} | \textit{CENTER} | \textit{LEFT} | \textit{RIGHT} | \textit{TOP} | \textit{TOPLEFT} | \textit{TOPRIGHT} | \textit{keyword-column}}

specifies the position of the text value with respect to the location of the data point.

\textbf{\textit{BOTTOM} | \textit{BOTTOMLEFT} | \textit{BOTTOMRIGHT} | \textit{CENTER} | \textit{LEFT} | \textit{RIGHT} | \textit{TOP} | \textit{TOPLEFT} | \textit{TOPRIGHT}}

specifies a position option.

\textit{keyword-column}

specifies a column that contains position options.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

The VCENTER= option specifies whether the position is relative to the text bound box or the text baseline. See the VCENTER= option. By default, the positions are relative to the text bounding box. The following figure shows the effect of each of these values on the position of an outlined text value when VCENTER=BBOX is in effect. The red dot indicates the marker data-point location.

| \textbf{POSITION=} When VCENTER=BBOX |
| \textbf{BOTTOM} | \textbf{BOTTOMLEFT} | \textbf{BOTTOMRIGHT} |
| \textit{Text} | \textit{Text} | \textit{Text} |
| \textbf{CENTER} | \textbf{LEFT} | \textbf{RIGHT} |
| \textit{Text} | \textit{Text} | \textit{Text} |
| \textbf{TOP} | \textbf{TOPLEFT} | \textbf{TOPRIGHT} |
| \textit{Text} | \textit{Text} | \textit{Text} |

When CENTER, LEFT, or RIGHT is specified, and VCENTER=BASELINE is in effect, the positions are relative to the text baseline, as shown in the following figure.
POSITION= When VCENTER=BASELINE

<table>
<thead>
<tr>
<th>CENTER</th>
<th>LEFT</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>Text</td>
<td>Text</td>
</tr>
</tbody>
</table>

Default CENTER

POSITIONOFFSETX=number | numeric-column
specifies the positive or negative X offset for an individual text item.

Note: This feature applies to SAS 9.4M5 and to later releases.

The offset is relative to the item's data coordinate. The value can be greater than 1. The offset unit is the font height of the current text item, which varies according to the text's font size based on size response, and so on.

Notes
A value of 1 implies an X offset that is the same distance as the current text item's font height.

A reverse axis reverses the offset direction.

Tip This option is most useful when the text is placed around other objects in the graph.

POSITIONOFFSETY=number | numeric-column
specifies the positive or negative Y offset for an individual text item.

Note: This feature applies to SAS 9.4M5 and to later releases.

The offset is relative to the item's data coordinate. The value can be greater than 1. The offset unit is the font height of the current text item, which varies according to the text's font size based on size response, and so on.

Notes
A value of 1 implies an Y offset that is the same distance as the current text item's font height.

A reverse axis reverses the offset direction.

Tip This option is most useful when the text is placed around other objects in the graph.

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

Default FALSE

Restriction This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

Note In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the
primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

See “When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

REVERSECOLORMODEL=TRUE | FALSE
specifies whether to reverse the gradient (color ramp) that is defined by either the ODS style that is in effect or by the COLORMODEL= option.

Default FALSE

See COLORMODEL= “boolean ” on page 1479 for other Boolean values that you can use.

ROTATE=number | numeric-column
specifies the angle of rotation, in degrees, for the text values. Positive angles are measured in a counter-clockwise direction, and negative angles are measured in a clockwise direction. You can use an angle that exceeds 360 degrees in absolute value.

Default 0

ROLENNAME=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

(role-name-list)
a space-separated list of role-name = column pairs.

Example The following example assigns the column Obs to the user-defined role TIP:

ROLENNAME=(TIP1=OBS)

Default No user-defined roles

Requirement The role names that you choose must be unique and different from the predefined roles X, Y, GROUP, and COLORRESPONSE.

SIZEMAX=dimension
specifies the maximum font size for a text marker when a response variable is used to size the text-marker font. By default, the font size of the text values that are associated with the maximum response column value is set to the value specified by this option.

Default Three times the size specified in the GraphDataText style element for the maximum response column value marker.

Interactions The SIZERESPONSE= option must be specified for this option to have any effect.

The SIZEMAXRESPONSE= option specifies the response value at which the maximum font size for a text marker is reached. The font
size for all text values that exceed the SIZEMAXRESPONSE= value is set to the value specified in this option.

**Tips**

Use the SIZEMAXRESPONSE= option to specify the response value at which the maximum font size for a text marker is reached.

Use the SIZEMIN= option to specify the minimum font size for text markers.

**See**

"dimension" on page 1480

**SIZEMAXRESPONSE=numeric | scalar-numeric-expression**

specifies the response value that corresponds to the maximum font size for text markers.

**Default**

The maximum value in the response column specified in the SIZERESPONSE= option.

**Interaction**

The SIZERESPONSE= option must be specified for this option to have any effect.

**Note**

The font size for all text values that exceed the maximum response value is set to the value specified in the SIZEMAX= option.

**SIZEMIN=dimension**

specifies the minimum font size for text markers when a response variable is used to size the font for text values.

**Default**

The size specified in the GraphDataText style element for the minimum response column value marker.

**Interaction**

The SIZERESPONSE= option must be specified for this option to have any effect.

**Tip**

Use the SIZEMAX= option to specify the maximum text size.

**See**

"dimension" on page 1480

**SIZERESPONSE=numeric-column | numeric-expression**

specifies a response column that is used to determine the font size for each text value.

**Default**

The size specified in the GraphDataText style element for all text values.

**Notes**

When the column value for an observation is 0, the font size for the text value for that observation is set to the SIZEMIN= option value.

When the column value for an observation is negative or missing, the text value for that observation is not displayed in the text plot. However, that observation still contributes to the axis ranges, legend, and so on.

When all the column values are 0 or missing, this option is ignored. In that case, the default font size is used for all of the text values.

**Tip**

Use the SIZEMIN= and SIZEMAX= options to limit the minimum and maximum font size for the text values.

**SPLITCHAR="character-list"**

specifies one or more characters on which the text-marker text can be split.
When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When SPLITPOLICY=SPLIT and a text value collision is detected, the text-marker text is split on a specified split character only if a split is needed at that point in order to make the text fit. In that case, a split might not occur on every split character. When SPLITPOLICY=SPLITALWAYS, the text-marker text is split unconditionally on every occurrence of a split character. If the text-marker text does not contain any of the specified split characters, then the text is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

Default

A blank space

Requirements

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

splitchar="abc"

The SPLITPOLICY= option must specify SPLIT or SPLITALWAYS.

Interaction

The SPLITCHARDROP= option specifies whether the split characters are included in the displayed data label or are dropped.

Notes

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

SPLITCHARDROP=TRUE | FALSE

specifies whether the split characters are included in the text-marker text.

TRUE

drops a split character from the text-marker text when a split occurs at that character.

Split characters at which a split does not occur are left in place. The SPLITPOLICY= option determines where the text is split. When SPLITPOLICY=SPLIT, the text for each text value is split at a split character only where a split is needed to fit the text in the available space. At each split point, the split character is dropped, and the characters that follow the split character, up to but not including the split character at the next split point, are wrapped to the following line.

When SPLITPOLICY=SPLITALWAYS, the text-marker text is split at every instance of a split character. All of the split characters are dropped. The characters that follow each split character, up to but not including the next split character, are wrapped to the next line.

FALSE

includes the split characters in the data label display.
The SPLITPOLICY= option determines how the split characters are displayed. When SPLITPOLICY=SPLIT, each data label is split at a split character only where a split is needed in order to make the label fit the available space. A split might not occur at every split character in the label. At each split point, the split character remains as the last character in the current line. The characters that follow the split character, up to and including the split character at the next split point, are then wrapped to the following line. This process repeats until all of the text is displayed.

When SPLITPOLICY=SPLITALWAYS, the text for each marker is split at every instance of a split character in the text regardless of whether a split is actually needed. Each split character remains as the last character in the current line. The characters that follow each split character, up to and including the next split character, are then wrapped to the next line.

Default: TRUE. A split character is dropped from the text-marker text when a split occurs at that character.

Requirement: The SPLITPOLICY= option must specify SPLIT or SPLITALWAYS.

See “boolean” on page 1479 for other Boolean values that you can use.

SPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT

specifies the justification of the lines of text in the text-marker text blocks.

AUTO

justifies the text based on the POSITION= option.

<table>
<thead>
<tr>
<th>POSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

CENTER

LEFT

RIGHT

justifies the text center, left, or right, as specified.

Default: AUTO

SPLITPOLICY=NONE | SPLIT | SPLITALWAYS

specifies a policy for avoiding collisions among the text values.

NONE

does not split the text for text values that collide.

SPLIT

splits the text-marker text at a split character only if a split is needed at that character in order to make the text fit the available space.

No split occurs at split characters that occur where a split is not needed. If the text does not contain any of the specified split characters, then a split does not occur. In that case, if the text does not fit the available space, then it might collide with the adjoining text values.
See the SPLITCHAR= option for information about specifying the split characters.

**SPLITALWAYS**

splits the text-marker text at every occurrence of a split character.

If the text does not contain any of the specified split characters, then a split does not occur.

See the SPLITCHAR= option for information about specifying the split characters.

Default: NONE

**SPLITWIDTH=** AUTO | width-in-characters

specifies the maximum width of each split line.

- **AUTO**
  uses the width of the longest inter-split-character substring.

- **width-in-characters**
  specifies a fixed width, expressed as a character count.

*Note: When you specify a fixed width, the text-marker text is split unconditionally every n characters, where n is the value of width-in-characters.*

Restriction: This option has an effect only when SPLITPOLICY=SPLIT.

**STRIP=** TRUE | FALSE

specifies whether leading and trailing blanks should be stripped from the text-marker text before it is displayed.

Default: FALSE

*Tip: Stripping the blanks from the numeric value strings helps center each string relative to its data point.*

See “boolean” on page 1479 for other Boolean values that you can use.

**TEXTATTRS=** style-element | style-element (text-options) | (text-options)

specifies the color and font properties of the text-marker text.

*Defaults:
  For non-grouped data, the GraphDataText style element.
  For grouped data, the Font attribute of the GraphDataText style element, and the ContrastColor attribute of a GraphDataN style element.*

*Interactions:
  When this option’s COLOR= suboption is used with the GROUP= option, the COLOR= suboption specifies the color for all of the text values.
  This option’s COLOR= suboption overrides the COLORRESPONSE= option. In that case, if a continuous legend is requested for the plot, the legend is not drawn.*

*Note:
  If one or more text options are specified and they do not include all the font properties (color, family, size, weight, and style), the
properties that are not specified are derived from the GraphDataText style element.

See
“General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element.

“Text Options” on page 1525 for available text-options.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over the text values. If you use this option, it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the scatter plot can be specified along with roles that do.

(role-list)
specifies an ordered, space-separated list of unique TEXTPLOT roles and user-defined roles. TEXTPLOT roles include X, Y, GROUP, and COLORRESPONSE.

User-defined roles are defined with the ROLENAME= option.

Example
The following example displays data tips for the columns assigned to the roles X and Y, as well as the column Obs, which is not assigned to any pre-defined TEXTPLOT role. The Obs column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y)
```

NONE
suppresses data tips and URLs (if requested) from the plot.

Default
The columns assigned to the following roles are automatically included in the data tip information: X, Y, GROUP, and COLORRESPONSE.

Restriction
Data tips in text plots are not supported in the SVG output format. To generate a text plot with data tips, use a bitmapped output format such as PNG.

Requirement
To generate data tips in the output, you must include an ODS GRAPHICS ON statement with the IMAGEMAP option specified, and you must write the output to the ODS HTML destination.

Interaction
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip
You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.
**Example**

ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

**Default**
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**TIPLABEL=(role-label-list)**
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

*role-label-list*

a space-separated list of \texttt{rolename} ="\texttt{string}" pairs.

**Example**

ROLENAME=(TIP1=PCT)
TIP=(TIP1)
TIPLABEL=(TIP1="Percent")

**Default**
The column label or column name of the column assigned to the role.

**Restriction**
Only the roles that appear in the TIP= option are used.

**Requirement**
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

**URL=string-column**
specifies an HTML page to display when a text value is selected.

*string-column*
specifies a column that contains a valid HTML page reference (HREF) for each text value that is an active link.

**Example**

\texttt{http://www.sas.com}

**Requirement**
To generate selectable text values, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Interactions**
This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tips**
The URL value can be blank for some X and Y pairs, meaning that no action is taken when the corresponding point is selected.

The URL value can be the same for any X and Y pair. In that case, the same action is taken when the points for that X and Y pair are selected.
By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

VCENTER=BBOX | BASELINE
specifies whether the text is vertically centered with respect to the text bounding box or to the text baseline.

BBOX
vertically centers the text with respect to its bounding box.

BASELINE
vertically centers the text with respect to the text baseline.

If the text is split into multiple lines, the text is centered on the baseline of the last line of text.

Restriction This option is valid only when POSITION= is set to CENTER, LEFT, or RIGHT. If POSITION= is set to any other value, VCENTER=BBOX is used instead.

Default BBOX

Tip Use the POSITION= option to specify the text position with respect to the text bounding box or to the text baseline.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default X

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default Y

Interaction The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

Details

About Text Plots

A text plot is basically a scatter plot that uses text as markers. To generate a text plot, you can use the SCATTERPLOT statement with the MARKERCHARACTER= option or you can use the TEXTPLOT statement. However, the TEXTPLOT statement provides more control over the marker text. With the TEXTPLOT statement, the plot data provides the marker location as X and Y coordinates, and the marker text. Each marker consists of the marker text enclosed in a bounding box.
that is centered on its X and Y coordinates. By default, the bounding box is not
visible. You can specify options in the TEXTPLOT statement to add an outline,
background fill, or both, to the bounding box. The following figure shows a simple
element of a text marker with a filled and outlined bounding box.

Using additional options in the TEXTPLOT statement, you can rotate the text around
its anchor point to any angle, manage the text position, split the text into multiple
lines, add a back-light effect to the text, and so on. If you currently use the
SCATTERPLOT statement and the MARKERCHARACTER= option to generate text
plots, SAS recommends that you use the TEXTPLOT statement instead.

Using Colors in a Text Plot

Text Marker Color Attributes

The TEXTPLOT statement enables you to control the following color attributes for
the text markers:

- text color
- bounding-box fill color
- bounding-box outline color

By default, only the marker text is displayed. Use the DISPLAY= option to specify
whether the bounding-box fill, bounding-box outline, or both are displayed.

The following table lists the options that you can use to control the color attributes.

<table>
<thead>
<tr>
<th>Option</th>
<th>Color Attribute</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXTATTRS=(COLOR=color)</td>
<td>text color</td>
<td>For non-grouped data, the GraphDataText:color style reference.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For grouped data, the GraphDataN:ContrastColor style reference.</td>
</tr>
<tr>
<td>FILLATTRS=(COLOR=color)</td>
<td>bounding-box fill color</td>
<td>For non-grouped data, the GraphDataDefault:color style reference.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For non-grouped data, the GraphDataN:color style reference.</td>
</tr>
<tr>
<td>OUTLINEATTRS=(COLOR=color)</td>
<td>bounding-box outline color</td>
<td>For non-grouped data, the GraphOutlines:contrastColor style reference.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For non-grouped data, the GraphDataN:contrastColor style reference.</td>
</tr>
</tbody>
</table>
When a response variable is used, in some cases, an attribute option is overridden by the response color. See “Response Colors in a Text Plot” on page 929.

Group Colors in a Text Plot

The TEXTPLOT statement supports the GROUP= option, which enables you to create a separate text marker for each unique value in a specified column. When the GROUP= option is in effect, by default, the text marker color attributes are controlled by the GraphData1–GraphDataN style elements. You can use the TEXTATTRS=, FILLATTRS=, and OUTLINEATTRS= options to set one or more color attributes to a fixed color. However, when you specify fixed colors, the contrast between the fixed colors and the remaining variable colors might not be sufficient for some group values. When both the GROUP= and COLORRESPONSE= options are in effect, the COLORRESPONSE= option controls the color attributes.

Response Colors in a Text Plot

The TEXTPLOT statement supports the COLORRESPONSE= option, which enables you to specify a numeric column or range attribute map variable that is used to determine the text-marker colors. When a numeric column is specified, each unique value is assigned a color from a color ramp that is specified in the COLORMODEL= option. When a range attribute map variable is specified, each unique value is assigned the attributes for that value that are defined in the attribute map. If the attribute map does not define the color attributes for a value, that value is assigned a color from the color ramp.

The TEXTPLOT statement COLORRESPONSE= option controls the color of the marker text, fill, and outline, depending on which bounding-box attributes are displayed. Prior to SAS 9.4M3, when the bounding-box fill and outline are displayed, COLORRESPONSE= determines the color of the text and the bounding-box fill. The bounding-box outline color in that case is determined by suboption COLOR= in the OUTLINEATTRS= option. When only the outline is displayed, the COLORRESPONSE= option determines the color of the text and the bounding-box outline. You can use suboption COLOR= in the FILLATTRS=, OUTLINEATTRS=, and TEXTATTRS= options to override the individual attributes.

Starting with SAS 9.4M3, when the bounding-box fill and outline are displayed, the COLORRESPONSE= option determines the color of the text and of the bounding-box outline. The bounding-box fill color in that case is determined by suboption COLOR= in the FILLATTRS= option. You can use suboption COLOR= in the FILLATTRS=, OUTLINEATTRS=, and TEXTATTRS= options to override the individual attributes, depending on which bounding-box attributes are displayed. The following table lists the results when suboption COLOR= is specified in some common color attribute option settings and the COLORRESPONSE= option is in effect.
The result depends on the bounding-box display as follows:

- If the bounding-box fill, outline, or both are displayed, the text color is set to its COLOR= specification. The bounding-box outline color is determined by the GraphDataDefault style element, and the bounding-box fill is assigned a color from the color ramp.
- If no bounding-box attributes are displayed, suboption COLOR= in TEXTATTRS= is ignored, and the text is assigned a color from the color ramp.

The result depends on the bounding-box display as follows:

- If the fill and outline are displayed, the text and fill colors are set to their COLOR= specification. The bounding-box outline is assigned a color from the color ramp.
- If only the fill is displayed, the text color is set to its COLOR= specification. Suboption COLOR= in FILLATTRS= is ignored, and the bounding-box fill is assigned a color from the color ramp.

The result depends on the bounding-box display as follows:

- If the fill and outline are displayed, the text and outline colors are set to their COLOR= specification. The bounding-box fill is assigned a color from the color ramp.
- If only the outline is displayed, the outline color is set to its COLOR= specification. Suboption COLOR= in TEXTATTRS= is ignored, and the text is assigned a color from the color ramp.

When the bounding-box fill and outline are displayed, suboption COLOR= in TEXTATTRS= is ignored, and the text is assigned a color from the color ramp. The fill and outline colors are set to their COLOR= specification.

### Customizing Text Marker Placement (Experimental)

Starting with **SAS 9.4M3**, you can use the TEXTPLOT statement `OUTFILE=` and `OUTID=` options to write information about the bounding box size and position for each text marker to a comma-separated values (CSV) file.

**CAUTION**

`OUTFILE=` and `OUTID=` are experimental options that are available in **SAS 9.4M3**. Do not use these options in production jobs.

These options are for users who want to customize the placement of the text in a text plot. After you write the attributes to a CSV file, you can import the data into a SAS data set, and then use the data to customize the placement of the text markers.

The following table lists the information that is written to the CSV file.
<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataDescent</td>
<td>Numeric</td>
<td>The descent, in y-axis units, of this observation’s text string. For a discrete axis, one unit is the distance between adjacent discrete category ticks.</td>
</tr>
<tr>
<td>DataHeight</td>
<td>Numeric</td>
<td>The height, in y-axis units, of this observation's text string. For a discrete axis, one unit is the distance between adjacent discrete category ticks.</td>
</tr>
<tr>
<td>DataWidth</td>
<td>Numeric</td>
<td>The width, in x-axis units, of this observation’s text string. For a discrete axis, one unit is the distance between adjacent discrete category ticks.</td>
</tr>
<tr>
<td>Descent</td>
<td>Numeric</td>
<td>The descent, in pixels, of this observation’s text string.</td>
</tr>
<tr>
<td>Height</td>
<td>Numeric</td>
<td>The height, in pixels, of the bounding box for this observation's text string.</td>
</tr>
<tr>
<td>Name</td>
<td>Character</td>
<td>The name that is specified in the NAME= option in this TEXTPLOT statement. The name associates this CSV file with a specific TEXTPLOT statement. This column is useful when multiple CSV files from different TEXTPLOT statements are merged into a single SAS data set.</td>
</tr>
<tr>
<td>OutID</td>
<td>Character</td>
<td>The value for this observation from the column specified in the OUTID= option in this TEXTPLOT statement. The ID is used as a unique identifier for this observation's text string.</td>
</tr>
<tr>
<td>RelativeDescent</td>
<td>Numeric</td>
<td>The relative descent of this observation's bounding box as a proportion of the plot area width.</td>
</tr>
<tr>
<td>RelativeHeight</td>
<td>Numeric</td>
<td>The relative height of this observation's text string as a proportion of the plot area height.</td>
</tr>
<tr>
<td>RelativeWidth</td>
<td>Numeric</td>
<td>The relative width of this observation's bounding box as a proportion of the plot area width.</td>
</tr>
<tr>
<td>Width</td>
<td>Numeric</td>
<td>The width, in pixels, of the bounding box for this observation's text string.</td>
</tr>
</tbody>
</table>

Here is an example of using the OUTFILE= and OUTID= options in a TEXTPLOT statement to write text bounding-box information to CSV file textboxdata.csv. It also shows how to import the CSV file into SAS data set Work.TextBoxData.

```sas
filename csvout "textboxdata.csv";
/* Write the text bound-box data to file textboxdata.csv */
proc template;
   define statgraph textplot;
```
begingraph;
  layout overlay;
  textplot x=weight y=height text=name / name="textboxdata"
    outfile=csvout outid=name;
  endlayout;
endgraph;
end;

proc sgrender data=sashelp.class template=textplot;
run;

/* Import the CSV file into data set Work.TextBoxData */
proc import
  out=work.textboxdata
  datafile=csvout
  dbms=csv replace;
  getnames=yes;
run;

Here is a partial listing of data set Work.TextBoxData.

<table>
<thead>
<tr>
<th>Obs</th>
<th>NAME</th>
<th>OUTID</th>
<th>WIDTH</th>
<th>HEIGHT</th>
<th>DESCENT</th>
<th>RELATIVEWIDTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>textboxdata</td>
<td>Alfred</td>
<td>34.045307</td>
<td>19.028384</td>
<td>5.004367</td>
<td>0.060364</td>
</tr>
<tr>
<td>2</td>
<td>textboxdata</td>
<td>Alice</td>
<td>31.040453</td>
<td>19.028384</td>
<td>5.004367</td>
<td>0.055036</td>
</tr>
<tr>
<td>3</td>
<td>textboxdata</td>
<td>Barbara</td>
<td>43.059872</td>
<td>19.028384</td>
<td>5.004367</td>
<td>0.076347</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obs</th>
<th>RELATIVEHEIGHT</th>
<th>RELATIVEDESCENT</th>
<th>DATAWIDTH</th>
<th>DATAHEIGHT</th>
<th>DATADESCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.046411</td>
<td>0.012206</td>
<td>6.489479</td>
<td>1.062499</td>
<td>0.279432</td>
</tr>
<tr>
<td>2</td>
<td>0.046411</td>
<td>0.012206</td>
<td>5.916715</td>
<td>1.062499</td>
<td>0.279432</td>
</tr>
<tr>
<td>3</td>
<td>0.046411</td>
<td>0.012206</td>
<td>8.207772</td>
<td>1.062499</td>
<td>0.279432</td>
</tr>
</tbody>
</table>

After you import the CSV file into a SAS data set, you can then use the data to position the text values as needed.

Example: TEXTPLOT Statement

This example creates a text plot of weight by age and sex. Column Name provides the text for filled, outlined markers. The following figure shows the output.
Here is the SAS code for this example.

```sas
proc template;
  define statgraph textplot;
  begingraph;
    entrytitle "Weight by Age and Sex";
    layout overlay / yaxisopts=(offsetmin=0.05 offsetmax=0.05);
    textplot x=age y=weight text=name / name='textplot1'
      display=all
      textattrs=(weight=bold) fillattrs=(transparency=0.9)
      group=sex groupdisplay=cluster clusterwidth=1;
    discretelegend 'textplot1';
  endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.class template=textplot;
run;
```

**VECTORPLOT Statement**

Creates a plot of vectors (directed line segments).

**Syntax**

```
VECTORPLOT X=numeric-column | expression
Y=numeric-column | expression
XORIGIN=numeric-constant | numeric-column | expression
YORIGIN=numeric-constant | numeric-column | expression <options>;
```
Summary of Optional Arguments

Appearance options

CLIP=TRUE | FALSE
specifies whether the origin is considered when determining the data ranges for the axes.

COLORMODEL=color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

COLORRESPONSE=numeric-column | range-attr-var | expression
specifies the column or range attribute variable to use to map the line colors to a continuous color gradient.

DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN
enhances the visual appearance of the vector plot lines.

DATATRANSPARENCY=number
specifies the degree of the transparency of the vector line and arrowhead, and the vector labels.

INDEX=positive-integer-column | expression
specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the vector line and arrowhead.

LINETHICKNESSMAX=dimension
specifies the maximum line thickness when a response variable is used to determine the line thickness.

LINETHICKNESSMAXRESPONSE=numeric | scalar-numeric-expression
specifies the response value that corresponds to the maximum line thickness.

LINETHICKNESSMIN=dimension
specifies the minimum line thickness when a response variable is used to determine the line thickness.

LINETHICKNESSRESPONSE=numeric-column | expression
specifies a response column or range attribute variable that is used to map a line thickness to each group value.

Axes options

PRIMARY=TRUE | FALSE
specifies that the data columns for this plot and the plot type be used for determining default axis features.

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

ROLENAME=(role-name-list)
specifies user-defined roles that can be used to display information in the data tips.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a vector line.
TIPFORMAT=(role-format-list)
   specifies display formats for tip columns.
TIPLABEL=(role-label-list)
   specifies display labels for tip columns.

Label options
   DATALABEL=column | expression
      specifies the labels at the ends of the vectors.
   DATALABELATTRS=style-element | style-element (text-options) | (text-options)
      specifies the color and font attributes of the data labels.
   DATALABELPOSITION=AUTO | TOPRIGHT | TOP | TOPLEFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
      specifies the location of the data labels relative to the end points and arrow heads.
   DATALABELSPLIT=TRUE | FALSE
      specifies whether to split the data labels at specified split characters.
   DATALABELSPLITCHAR="character-list"
      specifies one or more characters on which the data labels can be split if needed.
   DATALABELSPLITCHARDROP=TRUE | FALSE
      specifies whether the split characters are included in the data labels.
   DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
      specifies the justification of the strings that are inside the data label blocks.
   LEGENDLABEL="string"
      specifies a label to be used in a discrete legend for this plot.

Midpoint options
   GROUP=column | discrete-attr-var | expression
      creates a distinct set of vector lines and data label colors for each unique group value in the specified column.
   INCLUDEMISSINGGROUP=TRUE | FALSE
      specifies whether missing values of the group variable are included in the plot.

Plot reference options
   NAME="string"
      assigns a name to this plot statement for reference in other template statements.

Vector options
   ARROWDIRECTION=OUT | IN | BOTH
      specifies the placement of the arrowhead(s) at the end of the vector.
   ARROWHEADS=TRUE | FALSE
      specifies whether arrowheads are displayed on the vectors.
   ARROWHEADSHAPE=OPEN | CLOSED | FILLED | BARBED
      specifies the shape of the arrowheads.
   SCALE=number
      specifies the scale factor of the vector length.
Required Arguments

\textbf{X=numeric-column | expression}

specifies the column for the X values of the vector endpoints.

\textbf{Y=numeric-column | expression}

specifies the column for the Y values of the vector endpoints.

\textbf{XORIGIN=numeric-constant | numeric-column | expression}

specifies the X data coordinate of the vector origin.

\textbf{YORIGIN=numeric-constant | numeric-column | expression}

specifies the Y data coordinate of the vector origin.

Optional Arguments

\textbf{ARROWDIRECTION=OUT | IN | BOTH}

specifies the placement of the arrowhead(s) at the end of the vector.

\textbf{OUT}

specifies a single arrowhead, pointing away from the origin, at the end of
the vector away from the origin.

\textbf{IN}

specifies a single arrowhead, pointing toward the origin, at the end of
the vector near the origin.

\textbf{BOTH}

specifies two arrowheads.

One arrowhead points away from the origin, at the end of the vector opposite
from the origin. The other arrowhead points toward the origin, at the end of
the vector near the origin.

Default OUT

Interaction If \textbf{ARROWHEADS=} FALSE, then this option is ignored.

Tip Use the \textbf{ARROWHEADSHAPE=} option to control arrowhead
appearance.

\textbf{ARROWHEADS=} TRUE | FALSE

specifies whether arrowheads are displayed on the vectors.

Default TRUE

Interaction When this option is set to FALSE, the \textbf{ARROWDIRECTION=} and
\textbf{ARROWHEADSHAPE=} options are ignored and all vectors are
displayed as undirected line segments.

See “boolean ” on page 1479 for other Boolean values that you can use.

\textbf{ARROWHEADSHAPE=} OPEN | CLOSED | FILLED | BARBED

specifies the shape of the arrowheads.
Default: OPEN

Interaction: This option is ignored if ARROWHEADS= FALSE.

Note: No arrowhead is drawn for a zero-length vector. A zero-length vector is represented as a dot at its starting point.

Tip: Use the ARROWDIRECTION= option to control arrow direction.

**CLIP=**TRUE | FALSE
specifies whether the origin is considered when determining the data ranges for the axes.

**FALSE**
includes the origin when establishing the axis scales.

Each axis might be extended to force the display of the origin.

**TRUE**
ignores the origin when establishing axis scales.

Each axis scale is determined by the other plots in the overlay. This might result in the origin not being displayed if its data range is not within the data ranges of tips of the vectors.

Default: FALSE

See “boolean ” on page 1479 for other Boolean values that you can use.

**COLORMODEL=**color-ramp-style-element | (color-list)
specifies a color ramp to use with the COLORRESPONSE= option.

**color-ramp-style-element**
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

**STARTCOLOR**
specifies the color for the smallest data value of the COLORRESPONSE= column.

**NEUTRALCOLOR**
specifies the color for the midpoint of the range of the COLORRESPONSE= column.

**ENDCOLOR**
specifies the color for the highest data value of the COLORRESPONSE= column.

**color-list**
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement: The list of colors must be enclosed in parentheses.

See “color ” on page 1480

Default: The ThreeColorAltRamp style element.
Interaction  For this option to take effect, the COLORRESPONSE= option must also be specified.

**COLORRESPONSE=**numeric-column | range-attr-var | expression

specifies the column or range attribute variable to use to map the line colors to a continuous color gradient.

Note: This feature applies to **SAS 9.4M3** and to later releases.

**range-attr-var**
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“**RANGEATTRVAR Statement**”

Restriction  A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See  “**RANGEATTRVAR Statement**” on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Requirement  For a grouped plot, the COLORRESPONSE values should remain constant for each group value. If the COLORRESPONSE column has multiple values for a single GROUP value, unexpected results might occur.

Interactions  When the GROUP= option is specified with the COLORRESPONSE= option, the color attributes are controlled by the COLORRESPONSE= option.

Suboption **COLOR=** in the DATALABELATTRS= option overrides this option for the data label color attribute.

This option overrides suboption **COLOR=** in the LINEATTRS= option and varies the line color according to the color gradient or the attribute map.

Tips  To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

For a numeric column or expression, the ThreeColorAltRamp style element defines the line color gradient.

**DATALABEL=**column | expression

specifies the labels at the ends of the vectors.

Default  No data labels are displayed

Note  The label positions are automatically adjusted to prevent the labels from colliding with other labels and other arrows.
**DATALABELATTRS=** *style-element | style-element (text-options) | (text-options)*

specifies the color and font attributes of the data labels.

**Defaults**

For non-grouped data, the GraphDataText style element.

For grouped data, the GraphData1:ContrastColor–GraphDataN:ContrastColor style references.

**Interaction**

For this option to take effect, the **DATALABEL=** option must also be specified.

**Note**

When the **DATALABELPOSITION=AUTO** option is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Text Options” on page 1525 for available *text-options*.

**DATALABELPOSITION=** *AUTO | TOPRIGHT | TOP | TOpleFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT*

specifies the location of the data labels relative to the end points and arrow heads.

**AUTO**

uses a data-label collision avoidance algorithm to position the labels.

**See**

“LABELPLACEMENT=AUTO | GREEDY | SA” on page 33

**Note**

When **AUTO** is in effect, in some cases, the data label font size might be reduced in order to avoid overlapping labels and markers.

**TOPRIGHT | TOP | TOpleFT | LEFT | CENTER | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT**

specifies a location for all of the data labels.

**Default** AUTO

**DATALABELSPLIT=** *TRUE | FALSE*

specifies whether to split the data labels at specified split characters.

**Default** FALSE. The data labels are not split.

**Requirement**

The **DATALABEL=** option must also be specified.

**Interactions**

The **DATALABELSPLITCHAR=** option specifies one or more characters on which splits can occur.

This option has no effect when **DATALABELPOSITION=AUTO**.
See “boolean” on page 1479 for other Boolean values that you can use.

**DATALABELSPLITCHAR="character-list"**

specifies one or more characters on which the data labels can be split if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the data label. In that case, all of the specified split characters together are treated as a single split character.

When DATALABEL= is specified and DATALABELSPLIT=TRUE, the data label is split unconditionally at each occurrence of any of the specified split characters. If the data label does not contain any of the specified characters, then the label is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

**Default**

A blank space

**Requirements**

The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```plaintext
datalabelsplitchar="abc"
```

The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

**Interactions**

This option has no effect if DATALABELPOSITION=AUTO.

The DATALABELSPLITCHARDROP= option specifies whether the split characters are included in the data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the DATALABELSPLITJUSTIFY= option to specify the justification of the strings in the data label block.

**DATALABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the data labels.

**TRUE**

drops the split characters from the data label.

**FALSE**

includes the split characters in the data label.

When DATALABELSPLIT=TRUE and DATALABELSPLITCHARDROP=FALSE, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.
The following figure shows an example of a data label with the following specifications:

- The data label text for this label is Product*Group*A
- DATALABELSPLIT=TRUE
- DATALABELSPLITCHARDROP=TRUE | FALSE
- DATALABELSPLITCHAR="***"

When DATALABELSPLITCHARDROP=TRUE, the asterisks are removed from the label. When DATALABELSPLITCHARDROP=FALSE, each asterisk remains as the last character in the line prior to the new line.

Default: TRUE. The split characters are dropped from the data label.

Requirement: The DATALABEL= option and the DATALABELSPLIT=TRUE option must also be specified.

Interaction: The DATALABELSPLITCHAR= option specifies the split characters.

See: "boolean " on page 1479 for other Boolean values that you can use.

DATALABELSPLITJUSTIFY=AUTO | CENTER | LEFT | RIGHT
specifies the justification of the strings that are inside the data label blocks.

AUTO
justifies the labels based on the DATALABELPOSITION= option.

<table>
<thead>
<tr>
<th>DATALABELPOSITION= Value</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPLEFT, LEFT, or BOTTOMLEFT</td>
<td>RIGHT</td>
</tr>
<tr>
<td>TOPRIGHT, RIGHT, or BOTTOMRIGHT</td>
<td>LEFT</td>
</tr>
<tr>
<td>TOP, CENTER, or BOTTOM</td>
<td>CENTER</td>
</tr>
</tbody>
</table>

CENTER
LEFT
RIGHT
justifies the labels center, left, or right, as specified.

The following figure shows an example in which DATALABELPOSITION=TOP.

Note: The gray vertical line at the bottom of each label represents the horizontal center of the text box for reference.
In this case, because DATALABELPOSITION=TOP, AUTO centers the lines of text. The text box is anchored the same way that the unsplit text is anchored. For example, if DATALABELPOSITION=TOP, then the bottom center of the text box is positioned at the top of the marker.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**

enhances the visual appearance of the vector plot lines. The following figure shows a vector plot with each of the skins applied.

**Default**
The DATASKIN= option value that is specified in the BEGINGRAPH statement. If that value is not specified, then the GraphSkins:DataSkin style element value is used.

**Restriction**
Starting with **SAS 9.4M1**, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.

**Interaction**
This option overrides the BEGINGRAPH statement DATASKIN= option.

**DATATRANSPARENCY=number**
specifies the degree of the transparency of the vector line and arrowhead, and the vector labels.
GROUP=column | discrete-attr-var | expression
creates a distinct set of vector lines and data label colors for each unique group value in the specified column.

discrete-attr-var
specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction
A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Default
Each distinct group value might be represented in the plot by a different combination of color and line pattern. Lines vary according to the ContrastColor and LineStyle attributes of the GraphData1–GraphDataN and GraphMissing style elements.

Interactions
The group values are mapped in the order of the data, unless the INDEX= option is used to alter the default sequence of line patterns and colors.

The INCLUDEMISSINGGROUP= option controls whether missing group values are considered a distinct group value.

When both the GROUP= and the COLORRESPONSE= options are specified, the color attributes are controlled by the COLORRESPONSE= option.

Note
If you specify a column in a SAS data set, the visual attributes for each group value are assigned in data order. If you specify a column in a CAS in-memory table, the visual attributes for each group value are assigned in ascending order of the group column character values or of unformatted numeric values.

Tip
You can use the LINEATTRS= option to override the representations that are used to identify the groups. For example, You can use LINEATTRS=(PATTERN=SOLID) to assign the same pattern to all of the lines, letting the line color distinguish group values. Likewise, you can use LINEATTRS=(COLOR=BLACK) to assign the same color to all of the lines, letting the line pattern distinguish group values.

See
“DISCRETEATTRVAR Statement” on page 1435

INCLUDEMISSINGGROUP=TRUE | FALSE
specifies whether missing values of the group variable are included in the plot.

Default
TRUE

Interaction
For this option to take effect, the GROUP= option must also be specified.
Tip

The attributes of the missing group value are determined by the GraphMissing style element unless a discrete attribute map is in effect, the INDEX= option is used, the MISSING= system option changes the default missing character, or a user-defined format is applied to the group value. In those cases, the attributes of the missing group value are determined by a GraphData1–GraphDataN style element instead of by the GraphMissing style element.

See

“boolean” on page 1479 for other Boolean values that you can use.

**INDEX=positive-integer-column | expression**

specifies indices for mapping line attributes (color and line pattern) to one of the GraphData1–GraphDataN style elements.

**Requirements**

- The column or expression value must be an integer value of 1 or greater. Otherwise, this option is ignored.
- The positive-integer column must not contain missing values. Otherwise, the entire column is invalidated and this option is ignored.
- All of the indexes for a specific group value must be the same. Otherwise, the results are unpredictable.

**Interaction**

For this option to take effect, the GROUP= option must also be specified.

**Notes**

- The index values are 1-based indices. For the style attributes in GraphData1–GraphDataN, if the index value is greater than N, then a modulo operation remaps that index value to a number less than N to determine which style to use.
- If you do not use this option, then the group values are mapped in the order of the data.

**Tip**

You can use indexing to collapse the number of groups that are represented in a graph. For more information, see “Remapping Groups for Grouped Data” on page 198.

**LEGENDLABEL="string"**

specifies a label to be used in a discrete legend for this plot.

**Default**

The string specified on the NAME= option.

**Restriction**

This option applies only to an associated DISCRETELEGEND statement.

**Interaction**

If the GROUP= option is specified, then this option is ignored.

**LINEATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the vector line and arrowhead.

**Defaults**

- For non-grouped data, the GraphDataDefault style element.
- For grouped data, the ContrastColor, LineStyle, and LineThickness attributes of the GraphData1–GraphDataN style elements.
The COLORRESPONSE= option overrides this option’s COLOR= suboption.

The LINETHICKNESSRESPONSE= option overrides this option’s THICKNESS= suboption.

Note

The arrow head size is nonlinearly proportional to the line thickness in order to maintain appropriately sized arrow heads for thicker lines.

See

"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element

"Line Options" on page 1522 for available line-options.

LINETHICKNESSMAX=dimension

specifies the maximum line thickness when a response variable is used to determine the line thickness. By default, this option determines the thickness of the line that represents the maximum response column value.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default

Ten times the thickness that is specified by the GraphDataDefault style element LineThickness attribute.

Interactions

The LINETHICKNESSRESPONSE= option must be specified for this option to have any effect.

The LINETHICKNESSMAXRESPONSE= option specifies the response value at which this maximum line thickness is reached. The line thickness for response values that exceed the LINETHICKNESSMAXRESPONSE= value are set to the value that is specified by this option.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

Tip

Use the LINETHICKNESSMIN= option to specify the minimum line thickness.

See

"dimension" on page 1480

LINETHICKNESSMAXRESPONSE=numeric | scalar-numeric-expression

specifies the response value that corresponds to the maximum line thickness.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default

The maximum value in the response column that is specified in the LINETHICKNESSRESPONSE= option.

Interactions

The LINETHICKNESSRESPONSE= option must be specified for this option to have any effect.
The thickness for all lines that exceed the maximum response value is set to the value specified in the LINETHICKNESSMAX= option.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

**LINETHICKNESSMIN=** \texttt{dimension}

specifies the minimum line thickness when a response variable is used to determine the line thickness.

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**Default**

The thickness specified by the GraphDataDefault style element LineThickness attribute.

**Interactions**

The LINETHICKNESSRESPONSE= option must be specified for this option to have any effect.

If the line thickness that is calculated from the LINETHICKNESSMIN=, LINETHICKNESSMAX=, and LINETHICKNESSMAXRESPONSE= option values is less than 0.5 for a line, that line is not drawn.

**Tip**

Use the LINETHICKNESSMAX= option to specify the maximum line thickness.

**See**

"dimension" on page 1480

**LINETHICKNESSRESPONSE=** \texttt{numeric-column | expression}

specifies a response column or range attribute variable that is used to map a line thickness to each group value.

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**Default**

The GraphDataDefault style element LineThickness attribute.

**Interactions**

When the column values are all zero, all negative, or all missing, this option is ignored. In these cases, the default line thickness is used for all of the lines.

This option overrides suboption THICKNESS= in the LINEATTRS= option.

**Note**

The line thicknesses are not reflected in a discrete legend.

**NAME=** \texttt{"string"}

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**

The \texttt{string} is case sensitive, cannot contain spaces, and must define a unique name within the template.
This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**
The string is used as the default legend label if the LEGENDLABEL= option is not used.

**PRIMARY=TRUE | FALSE**
specifies that the data columns for this plot and the plot type be used for determining default axis features. This option is needed only when two or more plots within an overlay-type layout contribute to a common axis.

**Default**
FALSE

**Restriction**
This option is ignored if the plot is placed under a GRIDDED or LATTICE layout block.

**Note**
In an OVERLAY layout, only one plot in an overlay can be the primary plot on a per-axis basis. When no plot is designated as the primary plot, the first plot that can be a primary plot is considered the primary plot. If multiple plots specify PRIMARY=TRUE for the same axis, then the last such plot encountered is considered the primary plot.

**See**
“When Plots Share Data and a Common Axis” on page 979

“boolean ” on page 1479 for other Boolean values that you can use.

**ROLENAME=**(role-name-list)
specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

**Example**
The following example assigns the column Obs to the user-defined role TIP:

```
ROLENAME=(TIP1=OBS)
```

**Default**
No user-defined roles

**Requirement**
The role names that you choose must be unique and different from the predefined roles X, Y, DATALABEL, XORIGIN, YORIGIN, GROUP, and INDEX.

**SCALE=**number
specifies the scale factor of the vector length.

**Default**
1.0

**Restriction**
The number specified must be greater than 0.

**TIP=**(role-list) | NONE
specifies the information to display when the cursor is positioned over a vector line. If this option is used, then the information specified replaces all of the information that is displayed by default. You can specify roles for columns that do not contribute to the vector plot along with roles that do.
an ordered, space-separated list of unique VECTORPLOT roles and user-defined roles.

VECTORPLOT roles include X, Y, DATALABEL, XORIGIN, YORIGIN, GROUP, INDEX, and COLORRESPONSE.

Define user-defined roles with the ROLENAME= option.

Note  Starting with SAS 9.4M3, the COLORRESPONSE role is valid.

Example  The following example displays data tips for the columns assigned to the roles X, Y, GROUP, and the column Obs, which is not assigned to any pre-defined VECTORPLOT role. The Obs column must first be assigned a role.

```
ROLENAME=(TIP1=OBS)
TIP=(TIP1 X Y GROUP)
```

NONE  suppresses data tips from the plot.

Default  The columns assigned to these roles are automatically included in the data tip information: X, Y, DATALABEL, XORIGIN, YORIGIN, GROUP, and COLORRESPONSE.

Requirement  To generate data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction  This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip  You can control the labels and formats for the TIP roles with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)  specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

```
(role-format-list)
```

a space-separated list of role-name = format pairs.

Example  ROLENAME=(TIP1=SALARY)
TIP=(TIP1)
TIPFORMAT=(TIP1=DOLLAR12.)

Default  The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction  Only the roles that appear in the TIP= option are used.

Requirement  A column must be assigned to each of the specified roles. (See the ROLENAME= option.)
TIPLABEL=(role-label-list)
specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

role-label-list
a space-separated list of rolename = "string" pairs.

Example
ROLENAME=(TIP1=PCT)
TIP1=(TIP1)
TIPLABEL=(TIP1="Percent")

Default
The column label or column name of the column assigned to the role.

Restriction
Only the roles that appear in the TIP= option are used.

Requirement
A column must be assigned to each of the specified roles. (See the ROLENAME= option.)

XAXIS=X | X2
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default
X

Interaction
The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

YAXIS=Y | Y2
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default
Y

Interaction
The overall plot specification and the layout type determine the axis display. For more information, see "How Axis Features Are Determined" on page 973.

Details
By default in a VECTORPLOT, each vector starts at 0, 0 in the data space and is terminated with an open arrowhead. Zero-length vectors are represented by a dot at the starting point. To specify alternative coordinates for the starting point, use the XORIGIN= and YORIGIN= arguments.

Example: VECTORPLOT Statement
The following graph was generated by the "Example Program" on page 950:
Example Program

```sas
data CarPref;
  input Make $11. Model $13. (S1-S25) (1.);
datalines;
  Cadillac   Deville      8007990491240508971093809
  Chevrolet  Aveo         00512004234510430003515698
  Chevrolet  Cavalier     4053305814161643544747795
  Chevrolet  Malibu       6027400723121345545668658
  Dodge      Intrepid     7006000434101107333458708
  Dodge      Stratus      3005005635461302444675655
  Dodge      Neon         4005003641021602754476555
  Ford       Taurus       2024006715021443530648655
  Ford       Mustang      500719770502110850657555
  Ford       Focus        00210003030201500514078
  Honda      Accord       5956897609699952998975078
  Honda      Civic        4836709507488852567765075
  Lincoln    LS           7008990592230409962091909
  Pontiac    Firebird     0107895613201206958265907
  Volkswagen Jetta        485869650887795377895000
  Volkswagen Beetle       4858509709695795487885000
  Volvo      S40          998999890999987989919000
;
run;
* Compute Two Component Model;
ods graphics;
ods exclude all;
ods output mdpref=plotdata;
proc prinqual data=CarPref n=2 replace mdpref method=mgv;
  id model;
  transform monotone(S1-S25);
```

run;
ods select all;

proc template;
  define statgraph vectorplot;
  begingraph;
    entrytitle "Multidimensional Preference Analysis";
    entrytitle "of Preference Ratings for Automobiles";
    layout overlayequated / equatetype=fit cycleattrs=true;
    referenceline y=0 / datatransparency=0.7;
    referenceline x=0 / datatransparency=0.7;
    vectorplot y=vec2 x=vec1 xorigin=0 yorigin=0 /
      datalabel=label2var;
    scatterplot y=prin2 x=prin1 /
      datalabel=idlab1 primary=true
      markerattrs=(symbol=circlefilled);
    endlayout;
  endgraph;
end;
run;

proc sgrender data=plotdata template=vectorplot;
run;

WATERFALLCHART Statement

Creates a waterfall chart that is computed from input data.

Restriction: The WATERFALLCHART statement does not support CAS in-memory tables. In SAS 9.4M6 and earlier releases, unexpected results might occur when data stored in a CAS in-memory table is used. Starting with SAS Viya 3.5 and SAS Studio 5.2, the waterfall chart is dropped when data stored in a CAS in-memory table is used.

Interaction: A Waterfall chart accumulates response values in data order. Any change in the order of the X-axis values from the data order can adversely affect the waterfall chart. The X-axis value order can change when the Waterfall chart is overlaid with other plots or when it is used in a Lattice with uniform axes. It can also change when certain options are applied to the X axis.

Tip: Starting with SAS 9.4M3, you can use subpixel rendering with this statement. It is enabled by default. To disable subpixel rendering, specify SUBPIXEL=OFF in the BEGINGRAPH statement or in an ODS GRAPHICS statement. For information about the BEGINGRAPH statement SUBPIXEL= option, see SUBPIXEL= on page 35. For information about the ODS GRAPHICS statement SUBPIXEL= option, see "ODS GRAPHICS Statement" on page 1483.

Syntax

```
WATERFALLCHART CATEGORY=column | expression
RESPONSE=numeric-column | expression </options>;
```
Summary of Optional Arguments

Appearance options

**BARWIDTH=number**
specifies the width of a bar as a ratio of the maximum possible width.

**BASELINEATTRS=style-element | (line-options)**
specifies the appearance of the baseline.

**COLORGROUP=column | discrete-attr-var | expression**
specifies a column that is used to discretely color the transaction bars.

**COLORMODEL=color-ramp-style-element | (color-list)**
specifies a color ramp to use with the COLORRESPONSE= option.

**COLORRESPONSE=numeric-column | range-attr-var | expression**
specifies the numeric column or range attribute map variable that is used to determine the transaction-bar colors.

**CONNECTBREAK=TRUE | FALSE**
specifies whether the connect line is broken for values that have no observations.

**DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN**
enhances the visual appearance of the filled bars.

**DATATRANSPIRENCY=number**
specifies the degree of the transparency of the bar fill, bar outline, bar labels, and trend lines, if displayed.

**DISPLAY=(display-options) | STANDARD | ALL**
specifies which bar features to display.

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the filled transaction bars.

**FILLTYPE=SOLID | GRADIENT**
specifies the bar fill type.

**FINALBARATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the “final” bar, if displayed.

**INITIALBARATTRS=style-element | style-element (fill-options) | (fill-options)**
specifies the appearance of the “initial” bar, if displayed.

**INITIALBARVALUE=number**
specifies a value for the initial bar.

**OUTLINEATTRS=style-element | style-element (line-options) | (line-options)**
specifies the appearance of the bar outlines.

Axes options

**BASELINEINTERCEPT=number**
specifies the response axis intercept for the baseline.

**FINALBARTICKVALUE=“string”**
specifies a tick value to use on the category axis when the “final” bar is displayed.

**INITIALBARTICKVALUE=“string”**
specifies a tick value to use on the category axis when the “initial” bar is displayed.

**XAXIS=X | X2**
specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

**YAXIS=Y | Y2**
specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Data tip options

\textbf{ROLENAME}=(role-name-list)

specifies user-defined roles that can be used to display information in the data tips.

\textbf{TIP}=(role-list) | NONE

specifies the information to display when the cursor is positioned over a bar.

\textbf{TIPFORMAT}=(role-format-list)

specifies display formats for tip columns.

\textbf{TIPLABEL}=(role-label-list)

specifies display labels for tip columns.

Label options

\textbf{BARLABEL}=TRUE | FALSE

specifies whether the bar statistic value is displayed at the end of the bar.

\textbf{BARLABELATTRS}=style-element | style-element (text-options) | (text-options)

specifies the text properties of the bar label text.

\textbf{BARLABELFITPOLICY}=AUTO | NONE

specifies a policy for avoiding collisions among the bar labels when labels are displayed.

\textbf{BARLABELFORMAT}=format

specifies the text format used to display the bar label.

\textbf{LEGENDLABEL}="string"

specifies a label to be used in a discrete legend for this plot.

ODS options

\textbf{URL}=string-column

specifies an HTML page to display when a bar is selected.

Plot reference options

\textbf{NAME}="string"

assigns a name to this plot statement for reference in other template statements.

Statistics options

\textbf{COLORSTAT}=SUM | MEAN

specifies the statistic to use for computing the response colors.

\textbf{STAT}=SUM | MEAN

specifies the statistic to be computed for the RESPONSE axis.

Trend line options

\textbf{CONNECT}=START | END

determines whether trend lines connect to the adjacent bar’s starting or ending value.

\textbf{CONNECTATTRS}=style-element | style-element (line-options) | (line-options)

specifies the appearance of the trend lines that connect the bars.

\textbf{CONNECTDECREASINGATTRS}=style-element | style-element (line-options) | (line-options)
specifies the appearance of trend lines that denote a decreasing value between bars.

\[ \text{CONNECTINCREASINGATTRS}=\text{style-element} | \text{style-element (line-options)} | (\text{line-options}) \]

specifies the appearance of trend lines that denote an increasing value between bars.

**Required Arguments**

**CATEGORY=column | expression**
specifies the column or expression for the category values. Duplicated category values are summarized into a unique value. All values are treated as discrete.

**RESPONSE=numeric-column | expression**
specifies the numeric column or expression for the response values.

**Optional Arguments**

**BARLABEL=TRUE | FALSE**
specifies whether the bar statistic value is displayed at the end of the bar.

Default  \( \text{FALSE} \)

**Note**  By default, the bar-label format is derived from the format that is assigned to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

**Tip**  The font and color attributes for the label are specified by the BARLABELATRRS= option. The text format is specified by the BARLABELFORMAT= option.

**See**  “boolean” on page 1479 for other Boolean values that you can use.

**BARLABELATRRS=style-element | style-element (text-options) | (text-options)**
specifies the text properties of the bar label text.

Default  The GraphDataText style element.

**Requirement**  For this option to take effect, BARLABEL=TRUE must be specified.

**See**  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**BARLABELFITPOLICY=AUTO | NONE**
specifies a policy for avoiding collisions among the bar labels when labels are displayed.

**AUTO**
rotates the bar labels if the labels exceed the midpoint spacing.

If the labels collide horizontally due to thin bars, then AUTO drops all of the labels. The following figure shows an example.
See the `BARWIDTH=` option for more information about the bar spacing.

**NONE**

- does not rotate the bar labels.
- Labels that are too long overlap.

**Default**  
AUTO

**Requirement**  
For this option to take effect, `BARLABEL=TRUE` must be specified.

**BARLABELFORMAT=** *format*

specifies the text format used to display the bar label.

**Default**  
The bar-label format is derived from the format that is applied to the response column or from BEST6 if no format is assigned. The derived format retains the precision of the response-column format and, if necessary, increases the format width to accommodate the summarized value on the response axis.

**Requirement**  
For this option to take effect, `BARLABEL=TRUE` must be specified.

**Note**  
When a bar-label format is specified with this option, the bar labels are formatted as specified by `format`. The specified format is not automatically expanded to accommodate values that are too wide.

**Tip**  
If you want the bar-label format to expand automatically for summarized values on the response axis, specify the format for the response column rather than in this option.

**BARWIDTH=** *number*

specifies the width of a bar as a ratio of the maximum possible width.

**Default**  
0.85

**Range**  
0–1, where 0 is the minimum width, which is one pixel wide, and 1 is the maximum possible width

**Notes**  
This option is needed only to change the default behavior.
By default, the bar width automatically adjusts based on the number of bars to be displayed and the wall width.

Tip
To remove any inter-bar gap, set BARWIDTH=1.

**BASELINEATTRS=** *style-element | (line-options)*
specifies the appearance of the baseline.

**Default**
The GraphAxisLines style element.

**Notes**
The baseline is always drawn by default.

When *style-element* is specified, only the style element’s COLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip
To suppress the baseline, set the line thickness to 0:
baselineattrs=(thickness=0)

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Line Options” on page 1522 for available *line-options*.

**BASELINEINTERCEPT=** *number*
specifies the response axis intercept for the baseline. The baseline is always displayed in the chart, whether for a specified value or for the default value. When the BASELINEINTERCEPT= option is used, the axis range is adjusted to include the baseline, and the baseline is placed at the specified value on the response axis.

![Baseline Intercepts Example](image)

**Default**
0
Interactions

The value set by this option affects only the chart’s initial and final bars. If no initial bar value is specified, then the first transaction bar is drawn from 0, no matter what is set for the baseline value.

If necessary, the response axis data range is extended to include the baseline intercept. When a logarithmic response axis is requested and BASELINEINTERCEPT= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set BASELINEINTERCEPT= to a positive value.

Note

Label positions are automatically adjusted to prevent the labels from overlapping.

Tips

Control the appearance of the baseline with the BASELINEATTRS= option.

To suppress the baseline prior to SAS 9.4M5, use the BASELINEATTRS= option to set the line thickness to 0. Starting with SAS 9.4M5, specify DISPLAYBASELINE=OFF.

COLORGROUP=column | discrete-attr-var | expression

specifies a column that is used to discretely color the transaction bars.

discrete-attr-var

specifies a discrete attribute map variable.

The discrete attribute map variable is created by a DISCRETEATTRVAR statement.

Restriction

A discrete attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set by a dynamic variable.

Interactions

This option is ignored if the COLORRESPONSE= option is also specified.

If a column or expression is specified, then the unique column values are found and the transaction bar colors are derived from the GraphData1–GraphDataN style elements. The COLOR attribute is used for the bar fill colors and the CONTRASTCOLOR attribute is used for the bar outline colors.

If a variable that is associated with an attribute map is specified, then the color mapping defined by the associated DISCRETEATTRMAP statement is used for the transaction bars.

Notes

All of the COLORGROUP values for a specific category value must be the same. Otherwise, the results are unpredictable.

All the transaction bars have only one fill and one outline color, determined by the ODS style or set by the FILLATTRS= and OUTLINEATTRS= options.

If you specify a column in a SAS data set, colors are mapped to group values in data order. If you specify a column in a CAS in-memory table, colors are mapped to group values in ascending order of the group column character values or unformatted numeric values.
Tip To manage the color of the “initial” bar, use the INITIALBARATTRS= option. To manage the color of the “final” bar, use the FINALBARATTRS= option.

See “DISCRETEATTRMAP Statement” on page 1425
“DISCRETEATTRVAR Statement” on page 1435

**COLORMODEL=**

**color-ramp-style-element | (color-list)**
specifies a color ramp to use with the COLORRESPONSE= option.

**color-ramp-style-element**
specifies the name of a color-ramp style element.

The style element should contain these style attributes:

**STARTCOLOR**
specifies the color for the smallest data value of the COLORRESPONSE= column.

**NEUTRALCOLOR**
specifies the color for the midpoint of the range of the COLORRESPONSE= column.

**ENDCOLOR**
specifies the color for the highest data value of the COLORRESPONSE= column.

**color-list**
specifies a space-separated list of colors to use in the color ramp. You can use style attribute references such as GraphData3:Color, color names, or RGB, CMYK, HLS, and HSV (HSB) color codes to specify a color. The list can contain a mix of style attribute references, color names, and color codes.

Requirement The list of colors must be enclosed in parentheses.

See “color ” on page 1480

Default The ThreeColorRamp style element

Interactions For this option to take effect, the COLORRESPONSE= option must also be specified.

When FILLTYPE=GRADIENT and a color list is specified, the middle color in the list is treated as the NEUTRAL color.

Tip To manage the color of the initial bar, use the INITIALBARATTRS= option. To manage the color of the final bar, use the FINALBARATTRS= option.

**COLORRESPONSE=**

**numeric-column | range-attr-var | expression**
specifies the numeric column or range attribute map variable that is used to determine the transaction-bar colors.

**range-attr-var**
specifies a range attribute map variable that is defined in a RANGEATTRVAR statement.

“RANGEATTRVAR Statement”
Restriction  A range attribute map variable specification must be a direct reference to the attribute map variable. It cannot be set as a dynamic variable.

See  "RANGEATTRVAR Statement" on page 1448

When a numeric column or expression is specified, the range of column or expression values are linearly mapped to the color ramp that is specified by the COLORMODEL= option. When a range attribute map variable is specified, the colors that are defined in the associated range attribute map are used instead.

Restriction  This option affects only the fill colors. When only the bar outlines are displayed, this option has no effect.

Interactions  When the COLORGROUP= option is specified with this option, the COLORGROUP= option is ignored.

This option overrides suboption COLOR= in the FILLATTRS= option.

Tips  To display a legend with this option in effect, use a CONTINUOUSLEGEND statement.

Use the COLORSTAT= option to specify the statistic to compute for the COLORRESPONSE= column.

To produce discrete color mapping, the RANGEATTRMAP statement can define an attribute map that maps a single color to all values greater than 0, and another color to all values less than 0.

Use the FILLTYPE= option to indicate whether the color mapping is used to produce solid or gradient fills. When FILLTYPE=GRADIENT is in effect, the color at the end of the bar is based on the color mapping, and the neutral color of the color ramp is used as the starting color of each bar.

| COLORSTAT=SUM | MEAN |
|----------------|
| specifies the statistic to use for computing the response colors. |

Note: This feature applies to SAS 9.4M3 and to later releases.

Default  SUM

Interaction  This option is ignored when the COLORRESPONSE= option is not specified.

Tip  This option might affect existing SAS programs. For programs written before SAS 9.4M3, if STAT= and COLORRESPONSE= are specified in a WATERFALLCHART statement, the chart colors and color statistic might change from the previous SAS releases. To restore the original colors and color statistic in that case, set COLORSTAT= in the WATERFALLCHART statement to the same statistic that is specified in STAT=.
CONNECT=START | END
determines whether trend lines connect to the adjacent bar’s starting or ending value.

START
draws the trend lines horizontally and connects each to the adjacent bar’s starting value.
Each connecting line extends from the right corner of one bar’s ending value to the left corner of the adjacent bar’s starting value.

END
draws the trend lines diagonally and connects each to the adjacent bar’s ending value.
Each connecting line extends from the right corner of one bar’s ending value to the left corner of the adjacent bar’s ending value.

Default: START

Restriction: The last connect line is always drawn horizontally, extending from the right corner of the last data bar’s ending value to the left corner of the “final” bar’s starting value.

CONNECTATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the trend lines that connect the bars.

Default: The GraphConnectLine style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

CONNECTBREAK=TRUE | FALSE
specifies whether the connect line is broken for values that have no observations.

Note: This feature applies to SAS 9.4M5 and to later releases.

Default: FALSE

Requirement: DISPLAY= must include CONNECT for this option to have any effect.

Interaction: This option is ignored when the GROUP= option is in effect.

CONNECTDECREASINGATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of trend lines that denote a decreasing value between bars.

Default: The appearance specified by the CONNECTATTRS= option.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.
CONNECTINCREASINGATTRS=\textit{style-element} | \textit{style-element (line-options)} | (\textit{line-options})
specifies the appearance of trend lines that denote an increasing value between bars.

\begin{itemize}
  \item \textbf{Default}: The appearance specified by the CONNECTATTRS= option.
  \item \textbf{See} “General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}.
  \item “Line Options” on page 1522 for available \textit{line-options}.
\end{itemize}

\textbf{DATASKIN=NONE | CRISP | GLOSS | MATTE | PRESSED | SHEEN}
enhances the visual appearance of the filled bars. The following figure shows bars with each of the skins applied.

\begin{itemize}
  \item \textbf{Default}: The \textbf{DATASKIN=} option value that is specified in the \textbf{BEGINGRAPH} statement. If not specified, then the GraphSkins:DataSkin style element value is used.
  \item \textbf{Restriction}: Starting with \textbf{SAS 9.4M1}, the maximum number of skinned graphical elements is limited to 200 per plot in an OVERLAY or PROTOTYPE layout. When this limit is exceeded for a plot, the specified data skin is not applied to that plot. In that case, use the DATASKINMAX= option in your ODS GRAPHICS statement to increase the maximum limit.
  \item \textbf{Requirement}: For this option to have any effect, the fill must be enabled by the ODS style or the \textbf{DISPLAY=} option.
  \item \textbf{Interactions}: This option overrides the \textbf{BEGINGRAPH} statement \textbf{DATASKIN=} option.
  \begin{itemize}
    \item The skin appearance is based on the FILLATTRS= color.
    \item When a data skin is applied, all bar outlines are set by the skin, and the OUTLINEATTRS= option is ignored.
  \end{itemize}
\end{itemize}

\textbf{DATATRANSPARENCY=}\textit{number}
specifies the degree of the transparency of the bar fill, bar outline, bar labels, and trend lines, if displayed.
Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

Tip The FILLATTRS= option can be used to set transparency for just the filled bar area. The INITIALBARATTRS= and FINALBARATTRS= options can be used to specify transparency for the initial and final bars. You can combine this option with FILLATTRS=, INITIALBARATTRS=, and FINALBARATTRS= to set one transparency for the bar outlines and trend lines but a different transparency for the bar fills. Example:

data.transparency=0.2 fillattrs={transparency=0.6}

DISPLAY=(display-options) | STANDARD | ALL

specifies which bar features to display.

(display-options)
a space-separated list of one or more display options enclosed in parentheses.

The list can include one or more of the following options:

CONNECT
displays line segments (trend lines) connecting adjacent bar. The connection point is determined by the CONNECT= option.

FILL
displays filled bars.

FINALBAR
displays the final bar.

OUTLINE
displays outlined bars.

STANDARD
specifies CONNECT, FILL, FINALBAR, and OUTLINE

ALL
currently the same as STANDARD

Default STANDARD

Tips To control the appearance of the bars, use the COLORMODEL=, FILLATTRS=, and OUTLINEATTRS= options.

To control the appearance of the trend lines, use the CONNECTATTRS=, CONNECTDECREASINGATTRS=, and CONNECTINCREASINGATTRS= options.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the appearance of the filled transaction bars.

Defaults If the COLORGROUP= option is not specified, then the GraphDataDefault:Color style reference.

If the COLORGROUP= option is specified, then the GraphData1:Color–GraphDataN:Color style references.
Interaction  This option’s color specification is ignored if either the 
\texttt{COLORMODEL=} or \texttt{COLORRESPONSE=} option is specified. The 
transparency specification is honored in that case.

Tip  The \texttt{DATATRANSPARENCY=} option sets the transparency for the 
bar fill, bar outline, and trend lines. You can combine this option with 
\texttt{DATATRANSPARENCY=} to set one transparency for the bar outlines 
and trend lines but a different transparency for the bar fills. Example: 
\texttt{datatransparency=0.2 fillattrs=(transparency=0.6)}

See  “General Syntax for Attribute Options” on page 1519 for the syntax 
on using a \textit{style-element}.

“Fill Color Options” on page 1520 for available \textit{fill-options}.

\textbf{FILLTYPE=\texttt{SOLID | GRADIENT}}
\begin{itemize}
  \item \texttt{SOLID}  fills each bar with the color that is assigned to that bar.
  \item \texttt{ALPHAGRADIENT}  fills each bar with a color and a transparency gradient that starts at the bar 
top with the initial transparency that is assigned to that bar, and ends with full 
transparency at the bar baseline.
\end{itemize}

Note: This feature applies to \texttt{SAS 9.4M2} and to later releases.

\textbf{SOLID}  
\begin{itemize}
  \item \texttt{ALPHAGRADIENT}  fills each bar with a color and a transparency gradient that starts at the bar 
top with the initial transparency that is assigned to that bar, and ends with full 
transparency at the bar baseline.
\end{itemize}

Note: This feature applies to \texttt{SAS 9.4M5} and to later releases.

Tips  The fill color is determined by a style element or by the \texttt{FILLATTRS=} 
option \texttt{COLOR=} suboption. The initial transparency is determined by 
the \texttt{DATATRANSPARENCY=} option or by the \texttt{FILLATTRS=} option 
\texttt{TRANSPARENCY=} suboption.

For grouped plots, use the \texttt{FILLATTRS=} option in a discrete attribute 
map to set the initial transparency in the gradients for specific values.

\textbf{COLORGRADIENT}  
\begin{itemize}
  \item \texttt{COLORGRADIENT}  fills each bar with a color gradient that starts with the fill color that is assigned 
to that bar and ends with the color that is specified in the \texttt{FILLENDCOLOR=} 
option.
\end{itemize}

Note: This feature applies to \texttt{SAS 9.4M5} and to later releases.

\textbf{Alias}  \texttt{GRADIENT}

Interaction  The \texttt{SHEEN} data skin cannot be used when 
\texttt{FILLTYPE=GRADIENT} is in effect. You can use one of the other 
data skins.
Tip Use the DATATRANSPARENCY= option or the FILLATRHS= option TRANSPARENCY= suboption to set the initial transparency in the gradients.

See “FILLENDCOLOR=color” on page 576

**GRADIENT**

fills each bar with a color gradient that starts with the neutral color from the color model or with the middle color from the color model color list, if specified, and ends with the color for that bar that is determined by the COLORGROUP= or COLORRESPONSE= option.

Interaction The SHEEN data skin cannot be used when FILLTYPE=GRADIENT is in effect. You can use one of the other data skins.

Default SOLID

Interaction The DISPLAY= option must include FILL for this option to have any effect.

Tip The colors that are used depend on whether the COLORGROUP= option or the COLORRESPONSE= option is also specified.

See DATASKIN= on page 961

**FINALBARATTRS=style-element | style-element (fill-options) | (fill-options)** specifies the appearance of the “final” bar, if displayed.

Default the GraphFinal style element

Interaction This option is ignored if the DISPLAY= option does not display the “final” bar.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

**FINALBARTICKVALUE=“string”** specifies a tick value to use on the category axis when the “final” bar is displayed

Default “Final”

Interaction This option is ignored if the DISPLAY= option does not display the “final” bar.

**INITIALBARATTRS=style-element | style-element (fill-options) | (fill-options)** specifies the appearance of the “initial” bar, if displayed.

Default the GraphInitial style element

Interaction For this option to take effect, the INITIALBARVALUE= option must also be specified.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
“Fill Color Options” on page 1520 for available fill-options.

**INITIALBARTICKVALUE=**"string"

specifies a tick value to use on the category axis when the “initial” bar is displayed.

Default: "Initial"

Interaction: For this option to take effect, the INITIALBARVALUE= option must also be specified.

**INITIALBARVALUE=**number

specifies a value for the initial bar. The initial bar’s value is used as the starting response value for the first transaction bar.

If this option is not specified, then the initial bar is not included in the chart and the first transaction bar is drawn from response value 0. This is true even if an intercept value is set by the BASELINEINTERCEPT= option.

If necessary, the response axis data range is extended to include the initial bar value. When a logarithmic response axis is requested and INITIALBARVALUE= specifies 0 or a negative value, the response axis reverts to a linear axis. To restore the log axis in that case, set INITIALBARVALUE= to a positive value.

Note: The first transaction bar starts at response value 0.

**LEGENDLABEL=**"string"

specifies a label to be used in a discrete legend for this plot.

Default: The response-variable label. If a label is not defined, then the response-variable name is used.
<table>
<thead>
<tr>
<th><strong>Restriction</strong></th>
<th>This option applies only to an associated DISCRETELEGEND statement.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Interaction</strong></td>
<td>If the COLORGROUP= option is in effect, then this option is ignored.</td>
</tr>
</tbody>
</table>

**NAME="string"**

assigns a name to this plot statement for reference in other template statements. The specified name is used primarily in legend statements to coordinate the use of colors and line patterns between the plot and the legend.

**Restrictions**

The *string* is case sensitive, cannot contain spaces, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

**Interaction**

The *string* is used as the default legend label if the LEGENDLABEL= option is not used.

**OUTLINEATTRS=**

*style-element | style-element (line-options) | (line-options)*

specifies the appearance of the bar outlines.

**Default**

The ContrastColor and LineThickness attributes of the GraphOutlines style element.

**Interactions**

For this option to have any effect, outlines must be enabled by the ODS style or the DISPLAY= option.

If the DATASKIN= option applies a data skin, then this option is ignored.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Line Options” on page 1522 for available *line-options*.

**ROLENAME=(role-name-list)**

specifies user-defined roles that can be used to display information in the data tips. This option provides a way to add to the data columns that appear in data tips that are specified by the TIP= option.

**Example**

The following example assigns the column Obs to the user-defined role TIP:

ROLENANE=(TIP1=OBS)

**Default**

No user-defined roles

**Requirement**

The role names that you choose must be unique and different from the predefined roles CATEGORY, RESPONSE, COLORGROUP, and COLORRESPONSE.

**STAT=**

**SUM | MEAN**

specifies the statistic to be computed for the RESPONSE axis.

**Default**

SUM
Tip
If you use this option with COLORRESPONSE= in SAS programs that were written before SAS 9.4M3, the chart colors and color statistic might change from the previous SAS releases. To restore the original colors and color statistic in that case, set COLORSTAT= in the WATERFALLCHART statement to the same statistic that is specified in STAT=.

TIP=(role-list) | NONE
specifies the information to display when the cursor is positioned over a bar. If this option is used, then it replaces all of the information that is displayed by default. Roles for columns that do not contribute to the waterfall chart can be specified along with roles that do.

(role-list)
an ordered, space-separated list of unique WATERFALLCHART and user-defined roles.

WATERFALLCHART roles include CATEGORY, RESPONSE, COLORGROUP, and COLORRESPONSE.

Example
The following example displays data tips only for the column that is assigned to the RESPONSE role:

```
TIP=(RESPONSE)
```

NONE
suppresses data tips and URLs (if requested) from the plot.

Default
The columns assigned to these roles are automatically included in the data tip information: CATEGORY and RESPONSE.

Requirement
To enable data tips in the output, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Interaction
This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

Tip
The labels and formats for the TIP roles can be controlled with the TIPLABEL= and TIPFORMAT= options.

TIPFORMAT=(role-format-list)
specifies display formats for tip columns. This option provides a way to control the formats of columns that appear in data tips.

(role-format-list)
a space-separated list of role-name = format pairs.

Example
The following example displays data tips only for the column that is assigned to the RESPONSE role:

```
TIP=(RESPONSE)
TIPFORMAT=(RESPONSE=DOLLAR12.)
```

Default
The column format of the column assigned to the role or BEST6 if no format is assigned to a numeric column.

Restriction
Only the roles that appear in the TIP= option are used.
A column must be assigned to each of the specified roles.

**TIPLABEL=(role-label-list)**

specifies display labels for tip columns. This option provides a way to control the labels of columns that appear in data tips.

*role-label-list*

a space-separated list of rolename="string" pairs.

**Example**

TIP=(RESPONSE)

TIPLABEL=(RESPONSE="Average Sales")

**Default**

The column label or column name of the column assigned to the role.

**Restriction**

Only the roles that appear in the TIP= option are used.

**Requirement**

A column must be assigned to each of the specified roles.

**URL=string-column**

specifies an HTML page to display when a bar is selected.

*string-column*

specifies a column that contains a valid HTML page reference (HREF) for each bar that is to have an active link.

**Example**

http://www.sas.com

**Requirements**

To generate selectable bars, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

For non-grouped data, the values of the column are expected to be same for each unique RESPONSE value. If they are not, then the results might be unpredictable.

For grouped data, the values of the column are expected to be the same for each unique RESPONSE and group combination.

**Interactions**

This option has no effect when TIP=NONE.

This option is ignored when the plot statement is in an OVERLAY or PROTOTYPE layout and the INCLUDERANGES= option is specified in the LINEAROPTS= or TIMEOPTS= option for either axis.

**Tips**

The URL value can be blank for some RESPONSE values, meaning that no action is taken when the bars for those RESPONSE values are selected.

The URL value can be the same for different RESPONSE values, meaning that the same action is taken when the bars for those RESPONSE values are selected.

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target"
option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**

specifies whether data are mapped to the primary X (bottom) axis or to the secondary X2 (top) axis.

Default: X

**Interaction**
The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**YAXIS=Y | Y2**

specifies whether data are mapped to the primary Y (left) axis or to the secondary Y2 (right) axis.

Default: Y

**Interactions**
This option is ignored if the RESPONSE= argument is not specified.

The overall plot specification and the layout type determine the axis display. For more information, see “How Axis Features Are Determined” on page 973.

**Details**

The input data for the WATERFALLCHART statement is raw, unsummarized input data, and the statement calculates appropriate summarization statistics (sum or mean). By default, the bars in the chart appear in the order in which the CATEGORY values are present in the input data. A waterfall chart is typically used to show credit and debit transactions or successive changes to a given state.

In a waterfall chart, the bars that are calculated from the data are called “transaction” bars. The transaction bars represent the values of the RESPONSE variable across a series of intermediate values for the specified CATEGORY variable. You can manage the color of the transaction bars using the COLORGROUP, COLOREMODEL, or COLORRESPONSE= option.

A waterfall chart can also display an “initial” bar and a “final” bar. The value of the initial bar determines the starting response value for the first transaction bar. To set the initial value, use the INITIALBARVALUE= option. If the initial bar is not displayed, then the first transaction bar has a starting response value of 0. The value of the final bar is set automatically to the ending value of the last transaction bar.

**Example: WATERFALLCHART Statement**

The following graph was generated by the “Example Program” on page 970:
Example Program

```sas
data transactions;
input ID $ Amount type $;
datalines;
Alpha  2000 credit
Beta   -2500 debit
Gamma  -2000 debit
Delta  -500 debit
Epsilon  2250 credit
;
proc template;
define statgraph waterfallchart;
begingraph;
layout overlay;
  waterfallchart category=id response=amount / colorgroup=type
  initialbarvalue=1000
  name="waterfall";
discretelegend "waterfall";
endlayout;
endgraph;
end;
run;

proc sgrender data=transactions template=waterfallchart;
run;
```
Overview

GTL plots are specified within layout blocks that enable you to control the graph display features, including the display of the axes for the plots within the layout. For example, the LAYOUT OVERLAY statement has XAXISOPTS= and YAXISOPTS= options that enable you to specify axis features for the plots within the layout.

The following sections explain how the axis features are determined in a layout. The sections also discuss the issues that you must consider when setting the axis type or adjusting the appearance of the axis display.

How Axis Features Are Determined

Overview

GTL uses various criteria to determine the displayed axis features for a graph. Generally, axis features are based on the following criteria:

- the layout type
the order of plot statements in the layout and the options specified on those statements

- the use of primary and secondary axes on the plots (when secondary axes are supported)
- the plot type
- the column or columns of data that contribute to defining the axis range
- the data formats for the contributing data columns

Because the default axis features depend on a combination of the factors above, it is useful to understand how the axis features are determined in the templates that you build:

- how the data are mapped to the plot axes
- how the various layout types manage the axes.

---

**Plot Data Are Mapped to a Designated Axis**

### Overview for Axis Mapping

Depending on the layout type and the plots that you specify within the layout, you can manage up to four axes for two-dimensional plots:

- a primary X (bottom) axis
- a primary Y (left) axis
- a secondary X axis (X2 or “top” axis)
- a secondary Y axis (Y2 or “right” axis)

Within single-cell layouts (for example, OVERLAY layout), there can be just one each of these four axes. However, within multi-cell layouts (for example, LATTICE layouts), each cell can display the four axes. Thus, there can be multiple X, X2, Y, and Y2 axes across the columns and rows in the layout grid. In a lattice-type layout, you might have to use layout options to specify how the data ranges and axis display should be managed. This section discusses the simpler case for axis mapping, and "Axis Mapping in Lattice-type Layouts" on page 976 discusses the case for lattice-type layouts.

**Note:** GRIDDED layouts can be used to create a grid of cells, but the cells are independent. Thus, axes in the grid cannot be managed across columns and rows, so the simpler case applies to GRIDDED layouts.

---

**Primary and Secondary Axes**

By default, plot data are mapped to the primary axes. To enable you to override the default, plot statements that support a secondary X2 axis provide an XAXIS= option that can map data to the X or X2 axis. Plot statements that support a Y axis provide a YAXIS= option that can map data to the Y or Y2 axis.

To determine the axis features within a layout, GTL must first determine what data must be mapped to a particular axis. Thus, your use of primary and secondary axes...
on plot specifications affects GTL’s determination of default axis features for the layout.

For example, the plot statements in the following template specify Y-data mappings to the Y2 and Y axes:

```plaintext
proc template;
  define statgraph y2axis;
  begingraph;
    layout overlay;
      histogram weight / scale=count yaxis=y2;
      histogram weight / scale=percent yaxis=y;
      densityplot weight / normal();
    endlayout;
  endgraph;
end;
run;
proc sgrender data=sashelp.class template=y2axis;
run;
```

In this example, the first HISTOGRAM maps its Y-axis data to the Y2 axis, and the second HISTOGRAM maps its Y-axis data to the Y axis. The DENSITY plot does not explicitly map its Y-axis data, so the default Y axis is used. None of the plots explicitly maps X-data, so the default X axis is used for all three plots. Thus, GTL must manage any interactions that result from representing multiple plots on the X and Y axes. For example, on the X axis, it must determine an appropriate data range for representing the data values of all three plots.

When establishing axis features for each axis, GTL determines which plot specifications map data to the axis. GTL also collects the data for all of the plots that must be represented and maps that data to the designated axis. “When Plots Share Data and a Common Axis” on page 979 discusses the criteria GTL uses to determine the axis features for the axes after this mapping has been done for each axis.
Axis Mapping in Lattice-type Layouts

Lattice-type layouts (LAYOUT LATTICE, LAYOUT DATALATTICE, and LAYOUT DATAPANEL) present a grid of graphs that automatically aligns plot areas and tick display areas across grid cells. This alignment facilitates data comparisons among graphs, and for those comparisons to be meaningful, the graph axes must be coordinated across the columns and rows in the grid. All of the principles discussed in "Overview for Axis Mapping" on page 974 apply to the lattice-type layouts. In addition, because there can be multiple X, X2, Y, and Y2 axes across grid cells, you might have to use layout options to specify how the data ranges and axis display should be managed.

For example, the following template uses a LAYOUT LATTICE to generate a grid that displays a height analysis next to a weight analysis. By default in a LAYOUT LATTICE statement, the options ROWDATARANGE= and ROW2DATARANGE= are set to DATA. The DATA setting scales the Y-axis and Y2-axis data ranges separately for each cell in the layout. To ensure that the Y-axis data range is the same in both cells, the example specifies ROWDATARANGE=UNION. Similarly, to ensure that the Y2-axis data range is the same in both cells, the example specifies ROW2DATARANGE=UNION:

```sas
proc template;
  define statgraph y2axis;
  begingraph;
    layout lattice / columns=2 columngutter=10
      rowdatarange=union row2datarange=union
    ;
    layout overlay;
      histogram height / scale=count yaxis=y2;
      histogram height / scale=percent yaxis=y;
      densityplot height / normal();
    endlayout;
    layout overlay;
      histogram weight / scale=count yaxis=y2;
      histogram weight / scale=percent yaxis=y;
      densityplot weight / normal();
    endlayout;
  endlayout;
endgraph;
end;
proc sgrender data=sashelp.class template=y2axis;
run;
```
By default in a LAYOUT LATTICE statement, the options COLUMNDATARANGE= and COLUMN2DATARANGE= are also set to DATA. But in this analysis, the height is a separate measure from the weight, so the separate scales are appropriate for the X-axes across cells. If the X-axes were displaying the same measure (for example, comparing the height of female subjects to the height of male subjects), then you could specify COLUMNDATARANGE=UNIONALL. This would set the same scaling to the X-axis data ranges across the two layout columns. In this example, you would not bother changing the default COLUMN2DATARANGE= setting because the X2 axis is not needed.

Note: For DATALATTICE and DATAPANEL statements, UNIONALL is the default value for the data ranges. Thus, you would not have to change the data ranges unless you wanted to set UNION to scale data ranges per row or per column in the layout.

In the example, scaling the data ranges across the row ensures proper axis scaling. However, the graph display is cluttered by the duplicate display of ticks, axis values, and axis labels on both the Y and Y2 axes. To simplify the display, you can consolidate the axes. To do so, use a ROWAXES block to display a single Y axis for both cells, and a ROW2AXES block to display a single Y2 axis for both cells. The consolidated view removes the internal axes from the grid and displays only the external axes:

```plaintext
proc template;
  define statgraph y2axis;
  begingraph;
    layout lattice / columns=2 columngutter=10
      rowdatarange=union row2datarange=union;
      rowaxes;
      rowaxis / griddisplay=on;
    endrowaxes;
    row2axes;
    rowaxis;
  endgraph;
end;```

By default in a LAYOUT LATTICE statement, the options COLUMNDATARANGE= and COLUMN2DATARANGE= are also set to DATA. But in this analysis, the height is a separate measure from the weight, so the separate scales are appropriate for the X-axes across cells. If the X-axes were displaying the same measure (for example, comparing the height of female subjects to the height of male subjects), then you could specify COLUMNDATARANGE=UNIONALL. This would set the same scaling to the X-axis data ranges across the two layout columns. In this example, you would not bother changing the default COLUMN2DATARANGE= setting because the X2 axis is not needed.

Note: For DATALATTICE and DATAPANEL statements, UNIONALL is the default value for the data ranges. Thus, you would not have to change the data ranges unless you wanted to set UNION to scale data ranges per row or per column in the layout.

In the example, scaling the data ranges across the row ensures proper axis scaling. However, the graph display is cluttered by the duplicate display of ticks, axis values, and axis labels on both the Y and Y2 axes. To simplify the display, you can consolidate the axes. To do so, use a ROWAXES block to display a single Y axis for both cells, and a ROW2AXES block to display a single Y2 axis for both cells. The consolidated view removes the internal axes from the grid and displays only the external axes:

```plaintext
proc template;
  define statgraph y2axis;
  begingraph;
    layout lattice / columns=2 columngutter=10
      rowdatarange=union row2datarange=union;
      rowaxes;
      rowaxis / griddisplay=on;
    endrowaxes;
    row2axes;
    rowaxis;
  endgraph;
end;```
When using ROWAXES or ROW2AXES blocks in a LATTICE layout, you nest within the block one ROWAXIS statement for each row in the layout grid. The ROWAXIS statements are applied sequentially to the rows, and each ROWAXIS statement specifies the axis options for the Y or Y2 axes in its corresponding row. ROWAXIS statements within the ROWAXES block apply to the Y axes, and ROWAXIS statements within the ROW2AXES block apply to the Y2 axes. This example has just a single row in the grid, so each block specifies only one ROWAXIS statement. Notice that the ROWAXIS statement in the ROW2AXES block does not use any options. Thus, it consolidates Y2 axes in the row into a single, external Y2 axis, but it does not alter the default features of that axis. For columns in the grid, the LATTICE layout provides COLUMNAXES and COLUMN2AXES blocks. These blocks use COLUMNAXIS statements to externalize X and X2 axes and specify their features.

When you use DATALATTICE and DATAPANEL layouts, the layout dynamically generates a grid that contains as many cells as can be produced from the
combination of classification values. In those layouts the axes are always external, and you can use the COLUMNAXISOPTS=, COLUMN2AXISOPTS=, ROWAXISOPTS=, and ROW2AXISOPTS= options to specify the features for the axes. The settings on each option apply across the entire grid. For example, if you specify the ROWAXISOPTS= option in a DATALATTICE layout, then the specified settings apply to the external Y axes in every row.

When Plots Share Data and a Common Axis

Overview

If a layout block contains multiple plots that share data and a common axis, then the plot settings often interact in ways that affect the axis features. Axis features include the axis type, axis label, tick-mark layout, and so on. GTL resolves these interactions in ways that vary according to the layout block and plot statements.

Note: Axis interactions might not occur if other settings in the template prevent them. As discussed in “Plot Data Are Mapped to a Designated Axis” on page 974, if two plot statements are within an OVERLAY layout, then one of them might map its data to the X axis and the other might map its data to the X2 (top) axis. Mapping to separate axes can avoid the interactions that might occur if they both mapped their data to the X axis.

Axis Features in Overlay-type Layouts

Overlay-type layouts (OVERLAY, OVERLAYEQUATED, and PROTOTYPE, for example) build a composite from one or more GTL-statements. Within overlay-type layouts, if you do not explicitly set axis features in your template statements, then GTL automatically determines them. It sets the axis features based on the layouts and plots in the layout block and the data that are associated with the template at run time.

If only one plot statement within an overlay-type layout generates an axis, then determining axis features is straightforward: the features are derived directly from the plot type and the columns that are used for the plot data. For example, if a LAYOUT OVERLAY block contains a single SCATTERPLOT and the X= argument specifies a numeric column of children’s weights, then the default X-axis type is LINEAR. The default X-axis label is the column label of the Weight column. If the Weight column has no defined label, then the column name is used as a label.

When an overlay-type layout contains multiple plots that generate axes, GTL can determine default axis features for the shared axes. Alternatively, you can use the PRIMARY= option on one of the plot statements to specify which plot you want GTL to use. The following code fragment explicitly specifies that the SCATTERPLOT of children’s weights be used to determine axis features within the layout:

```
layout overlay;
  scatterplot x=weight ... / primary=true;
  ...
```

If no plot in an overlay-type layout is designated as primary, then the first plot that generates an axis is considered primary on a per-axis basis.
If PRIMARY=TRUE for a plot within an overlay-type layout, then that plot's data columns, data type, and plot type determine the default axis features. An explicitly specified primary plot determines the default axis features regardless of where that plot statement occurs within the layout block.

Only one plot can be primary on a per-axis basis. If multiple plots specify PRIMARY=TRUE for the same axis, then the last one encountered is considered primary.

The following SCATTERPLOT specifies a character column on the X= argument:

```plaintext
layout overlay;
  scatterplot x=name ... / primary=true;
...
```

In this case, the default X-axis type is DISCRETE and the X-axis label is the label that is assigned to column Name, or Name if no label is assigned to column Name.

Note: The SAS format on the primary plot's column determines the axis format, although the axis might not use that SAS format "as-is" from the column.

If a SCATTERPLOT's X= argument specifies a column that has a SAS DATETIME format, then the default X-axis type is TIME. The default X-axis label is the column label or name of the DateTime column:

```plaintext
layout overlay;
  scatterplot x=date ... / primary=true;
...
```

For some plot types, the default axis type does not directly correlate to the specified column's data type. For example, the following code fragment specifies a BARCHART for the numeric column Age:

```plaintext
layout overlay;
  barchart category=age ... / primary=true;
...
```

Because a BARCHART requires a discrete X axis, the default X-axis type in this case is DISCRETE, in spite of the fact that column Age is numeric. The X-axis label is the column label of Age, or the column name if no label exists.

Finally, consider a HISTOGRAM that is set as the primary plot in the layout and that bins data values:

```plaintext
layout overlay;
  histogram weight / binaxis=true primary=true;
...
```

In this case, the default X-axis type is LINEAR, but the histogram's data bins are used by default as the basis for the axis tick marks.

**Axis Features in Data Panel and Data Lattice Layouts**

The criteria discussed in "Axis Features in Overlay-type Layouts" on page 979 apply to determining the default axis features for the plots within DATAPANEL and DATALATTICE layouts. Both of these layout types nest a LAYOUT PROTOTYPE statement within their layout blocks. In both cases, the plot statements within the LAYOUT PROTOTYPE block—an overlay-type layout—determine the axis features for the plot display.
Axis Features in Lattice-type Layouts

The LAYOUT LATTICE statement can create a grid of graphs that automatically aligns plot areas, data display areas, labels, and headers across the columns and rows in the layout. The layout gives you the option of unifying the scale of the data ranges that are displayed in the graphs.

If a LAYOUT LATTICE specification generates only one cell, then no competition exists between cells for determining axis features in the display. In this case, the axis features are derived directly from the plot type and the columns used for the plot data.

Similarly, for multi-cell displays, if any or all of the options COLUMNDATARANGE=, COLUMN2DATARANGE=, ROWDATARANGE=, or ROW2DATARANGE= use the DATA setting to scale axis data ranges separately for each cell in the layout, then the layout cells are data-independent. The data-independent cells do not interact with each other for determining the axis features in the display.

Axes are shared in the layout when one of the options COLUMNDATARANGE=, COLUMN2DATARANGE=, ROWDATARANGE=, or ROW2DATARANGE= is used to unite axis data ranges for layout cells. By default in those cases, the first cell that is drawn (by default, the top left cell) determines the axis features in the display. When UNIONALL is in effect, those same features are used in all of the grid’s layout cells. When UNION is in effect, those same features are used on a per-row or per-column basis. If you specify external axes for the columns or rows in the layout, you can specify desired axis features on the appropriate COLUMNAXIS or ROWAXIS statements used in the layout.

For an example LAYOUT layout with external axes, see “Axis Mapping in Lattice-type Layouts” on page 976.

Axis Features in Gridded Layouts

In a GRIDDED layout the layout cells are independent of one another. Plot statements within the layout cells do not share data and are not represented on a common axis. Thus, no competition exists among layout cells for determining the axis features.

Plot Axis Types Must Agree on Common Axes

GTL is extremely flexible and enables you to generate a wide variety of plot displays. However, if you request incompatible plot displays within the same layout, then the results are unpredictable.

“When Plots Share Data and a Common Axis” on page 979 discusses the criteria GTL uses to determine the default axis features. After the axis type has been determined, GTL expects that all plots that share that axis will have the assigned axis type. The expectation applies whether you specify axis features in your template or let GTL determine default features.

For example, a BOXPLOT cannot be overlaid by a LINEPARM: the two types of plot cannot share axes because the plot types are incompatible within the same set of axes. Thus, if you were to use both a BOXPLOT statement and a LINEPARM statement within a LAYOUT OVERLAY block, then only one of them can be displayed. GTL therefore displays the primary plot (the first specified plot by default,
or the plot designated as primary by setting PRIMARY=TRUE). The other plot is not displayed.

Similarly, a BARCHART requires a discrete X axis, whereas a HISTOGRAM cannot be displayed on a discrete axis. If you specify both a BARCHART and a HISTOGRAM within the same overlay-type layout, then only the primary plot is displayed and the other plot is rejected from the display.

Axis types must also be the same for plots that must share an axis across the columns or rows in a multi-cell layout. For example, in a LAYOUT LATTICE, GTL expects that plots have the same axis type and data ranges if they are to share an external axis. Otherwise, the external axis cannot be displayed for that row or column.

### Controlling Axis Features

#### Overview

To enable you to control axis features within each of the layout types, there are different sets of axis options for the different types of axes:

<table>
<thead>
<tr>
<th>Option Category</th>
<th>Statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-dimensional axis options</td>
<td>LAYOUT OVERLAY statement</td>
</tr>
<tr>
<td>Three-dimensional axis options</td>
<td>LAYOUT OVERLAY3D statement</td>
</tr>
<tr>
<td>Equated axis options</td>
<td>LAYOUT OVERLAYEQUATED statement</td>
</tr>
<tr>
<td>Lattice axis options</td>
<td>LAYOUT LATTICE statement</td>
</tr>
<tr>
<td>DataLattice and DataPanel axes</td>
<td>LAYOUT DATALATTICE and LAYOUT DATAPANEL statements</td>
</tr>
</tbody>
</table>

The options that are available for each layout are documented separately, but it is worth discussing the following tasks in general for all of the layout types:

- Setting the Axis Type
- Adjusting the Axis View
- Adjusting Axis Thresholds
- Adjusting Axis Offsets
Setting the Axis Type

Within any given layout in the graph display, each plot axis is always of a particular type. In the default cases, the axis type is always LINEAR, DISCRETE, or TIME.

The axis options for each layout statement include a TYPE= option that enables you to specify an axis type that overrides the default selection mechanisms. When you override the default axis type, you must be sure to specify the correct axis type for the plots that you are defining. For every plot in the template language, the documentation indicates what axis types it supports. Plots statements that are specified in the template are ignored if they are incompatible with the axis type.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types:

- LINEAROPTS= (linear-suboptions)
- DISCRETEOPTS= (discrete-suboptions)
- TIMEOPTS= (time-suboptions)
- LOGOPTS= (log-suboptions)

One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

For example, a bar chart has two axes — a TYPE=DISCRETE axis for the X axis and a TYPE=LINEAR axis for the Y axis. If a numeric column (for example, Age) is assigned to the X role, then this column's values are always treated as discrete values, never as a continuous range of values. You cannot request another axis type for the X axis, but you can request a different axis type for the Y axis.

Sometimes you want a specialized axis type depending on the nature of the data. For example, if the data have a very large range of values (orders of magnitude apart), then you could request that the values be displayed on a logarithmic scale. To set a logarithmic scale, use the TYPE=LOG axis option.

Time series data benefit from displaying the X axis with a TYPE=TIME axis. A TIME axis type requires that the column values are SAS Date, Time, or Datetime values.

Three-dimensional plots such as BIHISTOGRAM3DPARM and SURFACEPLOTPARM always use TYPE=LINEAR for X, Y, and Z axes.

Note: Certain plot types or layouts might impose restrictions on what type of axis can be assigned. The documentation for each plot and layout type identifies any restrictions that might apply to the axes.

Adjusting the Axis View

The VIEWMIN= and VIEWMAX= axis options can be used to adjust the view of an axis. You can specify minimum data values to include in the display, maximum data values, or both (the specified values might be adjusted by the threshold calculation). By default, the VIEWMIN= value is the minimum data value for the specified axis and the VIEWMAX= value is the maximum data value for the specified axis.
A VIEWMIN= value that is greater than the data minimum or a VIEWMAX= value that is less than the data maximum acts like a “zoom in” operation. The adjusted view reduces the range of values represented on the axis and can sometimes exclude markers, lines, or fills that would normally appear.

A VIEWMIN= value that is less than the data minimum or a VIEWMAX= value that is greater than the data maximum acts like a “zoom out” operation. The adjusted view extends the range of values represented on the axis and sometimes compresses the markers, lines, or fills into a smaller area.

The following figure shows how the view settings can affect the tick and data displays.

---

### Adjusting Axis Thresholds

On a continuous, linear axis, the THRESHOLDMIN= and THRESHOLDMAX= axis options can be used to set a bias for including one more tick mark outside of either end of the data range (or VIEWMIN to VIEWMAX range). The threshold range is from 0 (do not include the tick mark) to 1 (include the tick mark). The default is 0.30. The bias at the minimum end of the axis is calculated using the THRESHOLDMIN= value and the minimum data value (by default) or the VIEWMIN= value (if set).

The bias at the maximum end of the axis is calculated using the THRESHOLDMAX= value and the maximum data value (by default) or the VIEWMAX= value (if set).

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range. Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

The following figure shows how the threshold settings can affect the tick display on an axis. In the figure, 8 is the minimum value for the display and 29 is the maximum value.
Adjusting Axis Offsets

The OFFSETMIN= and OFFSETMAX= axis options can be used to reserve an area at the minimum end of an axis, the maximum end, or both ends. No tick marks are displayed in the reserved areas.

The offset range is from 0 to 1, and the specified value is used to calculate the offset as a percentage of the full axis length. The larger the offset area that is reserved, the less space is available for the tick display area. The default offset reserves just enough area to fully display markers and other graphical features near the ends of an axis.

The following figure shows how offset values of 0.08 might compare with the default offsets for a continuous axis.
This next figure shows how offset values might affect the discrete axis of a bar chart.

![Diagram showing default offset and xaxisopts=(offsetmin=0.08 offsetmax=0.08)]
Axis Options in Layouts

Dictionary

Axis Options for LAYOUT OVERLAY

Axis options for the plots within an OVERLAY layout.

Interaction: The OVERLAY’s axis options are ignored when the LAYOUT OVERLAY statement is nested within another layout type that has external axes in effect. For example, the axis options are ignored when the OVERLAY is nested in a LAYOUT LATTICE with a COLUMNAXIS= or ROWAXIS= option in effect.

Note: Unless otherwise indicated in an option description, each axis option is available for the X, Y, X2, and Y2 axis.

See: “LAYOUT OVERLAY Statement” on page 149

Syntax

Axis options for the plots within an OVERLAY layout are specified with the following options on a LAYOUT OVERLAY statement:

\[ \text{XAXISOPTS} = \text{(axis-options)} \]
\[ \text{YAXISOPTS} = \text{(axis-options)} \]
\[ \text{X2AXISOPTS} = \text{(axis-options)} \]
\[ \text{Y2AXISOPTS} = \text{(axis-options)} \]

General Options for All Axes in an Overlay

The options that are documented in this section can be used with any of the axis types that are supported within an OVERLAY layout. Subsequent sections in the
chapter document the axis options that are available only for specific axis types: discrete, linear, log, or time axes. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Statement Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCRETEOPTS</td>
<td>Specifies options for a discrete axis.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Controls which axis features are displayed on the primary axis.</td>
</tr>
<tr>
<td>DISPLAYSECONDARY</td>
<td>Controls which axis features are displayed on the secondary axis.</td>
</tr>
<tr>
<td>GRIDATTRS</td>
<td>Specifies the attributes of the grid lines.</td>
</tr>
<tr>
<td>GRIDDISPLAY</td>
<td>Specifies whether axis grid lines are displayed.</td>
</tr>
<tr>
<td>LABEL</td>
<td>Specifies the axis label.</td>
</tr>
<tr>
<td>LABELATTRS</td>
<td>Specifies the color and font attributes of the axis label.</td>
</tr>
<tr>
<td>LABELFITPOLICY</td>
<td>Specifies a policy for fitting axis labels in the available space.</td>
</tr>
<tr>
<td>LABELPOSITION</td>
<td>Specifies the position of the axis label.</td>
</tr>
<tr>
<td>LABELSPLITCHAR</td>
<td>Specifies one or more characters on which the axis labels can be split, if needed.</td>
</tr>
<tr>
<td>LABELSPLITCHARDROP</td>
<td>Specifies whether the split characters should be included in the displayed axis labels.</td>
</tr>
<tr>
<td>LABELSPLITJUSTIFY</td>
<td>Specifies the justification of the strings that are inside the axis label blocks.</td>
</tr>
<tr>
<td>LINEAROPTS</td>
<td>Specifies features for a standard numeric interval axis.</td>
</tr>
<tr>
<td>LINEEXTENT</td>
<td>Specifies the extent of the axis line.</td>
</tr>
<tr>
<td>LOGOPTS</td>
<td>Specifies features for a log axis.</td>
</tr>
<tr>
<td>NAME</td>
<td>Assigns a name to an axis for reference in other statements.</td>
</tr>
<tr>
<td>OFFSETMAX</td>
<td>Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>OFFSETMIN</td>
<td>Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>Statement Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>REVERSE</td>
<td>Specifies whether the tick values should appear in the reverse order.</td>
</tr>
<tr>
<td>SHORTLABEL</td>
<td>Specifies an alternate axis label to use if the default or specified axis label is too long for the axis length.</td>
</tr>
<tr>
<td>TICKSTYLE</td>
<td>Specifies the placement of tick marks in relation to the axis line.</td>
</tr>
<tr>
<td>TICKVALUEATTRS</td>
<td>Specifies the color and font attributes of the axis tick values.</td>
</tr>
<tr>
<td>TICKVALUEHALIGN</td>
<td>Specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.</td>
</tr>
<tr>
<td>TICKVALUEVALIGN</td>
<td>Specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.</td>
</tr>
<tr>
<td>TIMEOPTS</td>
<td>Specifies features for a TIME axis.</td>
</tr>
<tr>
<td>TYPE</td>
<td>Specifies the type of axis to use.</td>
</tr>
</tbody>
</table>

**DISCRETEOPTS=(discrete-axis-options)**

specifies one or more options for a discrete axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

**Interaction**

This option is ignored if the axis type is not DISCRETE.

**See**

“Options for Discrete Axes Only” on page 1002 for the options that you can use for discrete-axis-options.

**DISPLAY=STANDARD | ALL | NONE | (display-options)**

controls which axis features are displayed on the primary axis.

**STANDARD**

specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

**ALL**

currently the same as STANDARD

**NONE**

specifies that no axis features are displayed

**display-options**

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

**LABEL**

displays the axis label

**LINE**

displays the axis line
TICKS
displays the tick marks

TICKVALUES
displays the values that are represented by the major tick marks

Default  STANDART

Tips  The default line attributes for the axis line and axis tick marks are
defined in the GraphAxisLine style element.

Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid
lines.

When LINE is excluded from the DISPLAY= option, the layout wall
outline or the default baseline of a bar chart, needle plot, or waterfall
chart can appear to be an axis line. To suppress the wall outline, use the
WALLDISPLAY= option in the layout statement. To suppress the plot
baseline, use the BASELINEATTRS= option in the plot statement.

DISPLAYSECONDARY=NONE | ALL | STANDART | (display-options)
controls which axis features are displayed on the secondary axis. A secondary
axis is not an independent axis. Rather, it mirrors the primary axis. Thus, for this
option to take effect, all plot statements in the layout must map data to the same
primary axis. For example, a secondary X2 axis can be displayed on top in the
layout, provided all plot statements set XAXIS=X to map data to the primary X
axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the
layout, provided all plot statements set YAXIS=Y to map data to the primary Y
axis (left).

NONE
specifies that no axis features are displayed

STANDART
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on
the secondary axis

ALL
currently the same as STANDART

(display-options)
specifies a space-separated list of one or more display options, enclosed in
parentheses.

The list can contain one or more of the following options:

LABEL
displays the axis label

LINE
displays the axis line

TICKS
displays the tick marks

TICKVALUES
displays the values that are represented by the major tick marks

Default  NONE

Restriction  If some plot statements set XAXIS=X and others set XAXIS=X2,
both the X and X2 axis are primary and a secondary X axis cannot
be displayed. In that case, this option is ignored. The same applies for the Y axes.

**Tip**
Use the `GRIDDISPLAY=` and `GRIDATTRS=` options to set the axis grid lines.

**GRIDATTRS=style-element | style-element (line-options) | (line-options)**
specifies the attributes of the grid lines.

**Default**
The GraphGridLines style element.

**Interaction**
This option is ignored if the `GRIDDISPLAY=` option does not display the grid lines.

**Tip**
On a log axis, this option affects the appearance of the major grid lines only. It does not affect the appearance of the minor grid lines. To control the appearance of the minor grid lines on a log axis, use the `MINORGRIDATTRS=` option.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Line Options” on page 1522 for available `line-options`.

**GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF**
specifies whether axis grid lines are displayed. This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

**AUTO_OFF**
specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains `DisplayOpts="ON."`

**AUTO_ON**
specifies that grid lines are displayed unless the GraphGridLines element in the current style contains `DisplayOpts="OFF."`

**ON**
specifies that grid lines are always displayed.
The current style has no override.

**OFF**
specifies that grid lines are never displayed.
The current style has no override.

The following table shows the end results for various combinations of the `GRIDDISPLAY=` option and the `DisplayOpts=` attribute of the GraphGridLines style element. Most supplied templates use the default setting `AUTO_OFF` to indicate a preference for not displaying grid lines, but allowing the style to override.

<table>
<thead>
<tr>
<th><code>GRIDDISPLAY=</code> option</th>
<th><code>DisplayOpts=</code> attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_OFF</td>
<td>AUTO</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>ON</td>
<td>yes</td>
</tr>
</tbody>
</table>
### GRIDDISPLAY= option

<table>
<thead>
<tr>
<th>GRIDDISPLAY= option</th>
<th>DisplayOpts= style attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_OFF</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>AUTO</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>ON</td>
<td>any value</td>
<td>yes</td>
</tr>
<tr>
<td>OFF</td>
<td>any value</td>
<td>no</td>
</tr>
</tbody>
</table>

**Default** AUTO_OFF

**Note** Supplied styles use DisplayOpts="AUTO," which means that the style has no preference about grid lines and the graphics template setting for grid lines is always used.

### LABEL="string" | ("string" ..."string")

specifies the axis label. The string can be either a string literal or a dynamic. The list form implies that all included string literals or dynamic variables will be concatenated.

**Default**

The default label is derived from the primary plot in the layout. For more information, see "When Plots Share Data and a Common Axis" on page 979.

**Interaction**

This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

**Note**

If the axis label is too long to fit along the axis, then it is truncated by default.

**Tips**

Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Use the ODS escape sequence to specify Unicode characters in an axis label. Here is an example that specifies degrees Fahrenheit in an axis label for temperature:

```
label="Temperature ((\*ESC\*)\{unicode '00B0'x\}F)"
```

**See**

"Overriding the Default Axis Label" in SAS Graph Template Language: User’s Guide

### LABELATTRS=style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the axis label.

**Default**

The GraphLabelText style element.
Interaction This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**LABELFITPOLICY=**AUTO | SPLIT | SPLITALWAYS | STACKED

specifies a policy for fitting axis labels in the available space.

**AUTO**
uses the short label, when specified, instead of the original label.

If the short label does not fit, then it is clipped. When no short label is specified, the original label is clipped.

**SPLIT**
splits the axis label at a split character only when necessary in order to make the label fit the available space.

The split character is specified by the LABELSPLITCHAR= option. The short label is not used. A split does not occur at a split character if a split is not needed at that location. If the label does not contain any of the specified split characters, then it is not split. A label that cannot be split or that does not fit the available space even after splitting might overlap the adjoining space.

**SPLITALWAYS**
always split the axis label at every occurrence of a split character, which is specified by the LABELSPLITCHAR= option.

If the label cannot be split, then it is clipped.

**STACKED**
displays the Y or Y2 axis label vertically as stacked letters.

**Note:** This feature applies to **SAS 9.4M5** and to later releases.

**Valid in** YAXISOPTS= and Y2AXISOPTS only

**Default** AUTO

**Interactions** This option has effect only when LABELPOSITION= is CENTER or DATACENTER.

When the overlay layout is nested in a LATTICE layout, SPLIT is ignored and AUTO is used instead.

**Note** When LABELPOSITION=CENTER, the available area is the full axis, including the axis offsets. When LABELPOSITION=DATACENTER, the available area is the tick display area, excluding the axis offsets.

**LABELPOSITION=**CENTER | DATACENTER | TOP | BOTTOM | LEFT | RIGHT

specifies the position of the axis label.

**CENTER**
centers the axis label in the axis area.
For the Y and Y2 axes, the label is oriented vertically and is centered in the axis area (including the offsets). The label is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis. For the X and X2 axes, the label is centered in the axis area (including the offsets). It is positioned below the tick values for the X axis or above the axis values for the X2 axis.

**DATACENTER**

centers the axis label in the axis tick display area.

For the Y and Y2 axes, the label is oriented vertically and is centered in the axis tick display area (excluding the offsets). It is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis. For the X and X2 axes, the label is centered in the axis tick display area (excluding the offsets). The label is positioned below the tick values for the X axis or above the axis values for the X2 axis.

**TOP**

orients the label horizontally at the top or bottom of the axis area.

The label is right-justified in the axis area for the Y axis and left-justified for the Y2 axis. If there is not sufficient room in the axis area to display the label, then the label grows to the right for the Y axis and to the left for the Y2 axis.

**BOTTOM**

orients the label horizontally at the top or bottom of the axis area.

When TOP or BOTTOM is used, the label might collide with other graphical features. In that case, use CENTER or DATACENTER instead.

**LEFT**

positions the label to the left or right of the axis area.

The label is centered vertically in the axis area.

**RIGHT**

positions the label to the left or right of the axis area.

The label is centered vertically in the axis area.

Restriction: These options are valid for the Y and Y2 axes only.

Note: When LEFT or RIGHT is used, the label might collide with other graphical features.

The following figure shows the CENTER and DATACENTER positions for a blue Y-axis label Type and a red X-axis label MPG.

<table>
<thead>
<tr>
<th>CENTER / CENTER</th>
<th>DATACENTER / DATACENTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>Hybrid</td>
</tr>
<tr>
<td>Truck</td>
<td>Truck</td>
</tr>
<tr>
<td>Wagon</td>
<td>Wagon</td>
</tr>
<tr>
<td>Sports</td>
<td>Sports</td>
</tr>
<tr>
<td>Sedan</td>
<td>Sedan</td>
</tr>
<tr>
<td>SUV</td>
<td>SUV</td>
</tr>
</tbody>
</table>

In this example, an axis offset is applied to the maximum end of both axes in order to demonstrate the difference between CENTER and DATACENTER. CENTER centers the labels on the entire axis area, including the offset.
DATACENTER centers the labels on the tick display areas, which does not include the offset.

The next figure shows the TOP and LEFT positions, and the BOTTOM and RIGHT positions for the same axis labels.

The following figure shows the CENTER and DATACENTER positions for a blue Y-axis label Qtr and a red X-axis label Close.

The next figure shows the TOP and LEFT positions, and the BOTTOM and RIGHT positions for the same axis labels.

Default CENTER

Restriction This option does not support collision avoidance. In some cases, axis label collisions can occur in the axis area.

Interaction When LEFT, RIGHT, TOP, or BOTTOM is in effect, the SHORTLABEL= option is ignored.

See SHORTLABEL= on page 999 for information about how short labels are used.

LABELSPLITCHAR="character-list"

specifies one or more characters on which the axis labels can be split, if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the axis label. In that case, all of the specified split characters together are treated as a single split character.

When LABELFITPOLICY=SPLIT, if the axis label does not fit the available space, then it is split on a specified split character only if a split is needed at that point to make the label fit. In this case, a split might not occur on every split character. When LABELFITPOLICY=SPLITALWAYS, the axis label is split unconditionally on every occurrence of a split character. If the axis label does not contain any of the specified split characters, the label is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

Default A blank space

Requirements The list of characters must be enclosed in quotation marks.
Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```plaintext
labelsplitchar="abc"
```

**Interactions**

This option has effect only when `LABELPOSITION=` is CENTER or DATACENTER.

The `LABELSPLITCHARDROP=` option specifies whether the split characters are included in the displayed data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the `LABELSPLITJUSTIFY=` option to specify the justification of the strings in the axis label block.

### `LABELSPLITCHARDROP=TRUE | FALSE`

specifies whether the split characters are included in the displayed axis labels.

- **TRUE**
  - drops the split characters from the axis label display.

- **FALSE**
  - includes the split characters in the axis label display.

  When the label is split while `LABELSPLITCHARDROP=FALSE` is in effect, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

**Default**

TRUE. The split characters are dropped from the axis label.

**Interactions**

This option has effect only when `LABELPOSITION=` is CENTER or DATACENTER.

The `LABELSPLITCHAR=` option specifies the split characters.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

### `LABELSPLITJUSTIFY=justification`

specifies the justification of the strings that are inside the axis label blocks.

**justification**

- **CENTER**
- **LEFT**
- **RIGHT**

  specifies the justification for the X or X2 axis label.

- **CENTER**
- **TOP**
- **BOTTOM**

  specifies the justification for the Y or Y2 axis label.

**Default**

CENTER
This option has effect only when `LABELPOSITION= CENTER` or `DATACENTER`.

**LINEAROPTS=(linear-axis-options)**

specifies one or more options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a `name = value` pair and each pair is space separated.

**Interaction**

This option is ignored if the axis type is not LINEAR.

**See**

“Options for Linear Axes Only” on page 1012 for the options that you can use for `linear-axis-options`.

**LINEEXXTENT=FULL | DATA | number**

specifies the extent of the axis line.

**Note:** This feature applies to **SAS 9.4M3** and to later releases.

**FULL**

specifies an axis line that extends along the entire length of the axis.

**DATA**

specifies an axis line that extends through the data range from the minimum offset to the maximum offset.

**number**

specifies how much the axis line extends from DATA toward FULL as a decimal proportion. A value of 0 is equivalent to DATA, and a value of 1 is equivalent to FULL.

**Range** 0–1

**Tip**

A numeric value is useful for bar charts when DATA terminates the axis line at the midpoint positions of the minimum and maximum bars. In that case, you can specify a numeric value to lengthen the axis line so that it extends to the full width of both bars.

The following figure shows a simple example of each value for the X axis and Y-axis lines. The light-blue dashed lines depict the minimum and maximum offsets that are set on the axes.

**Default** FULL

**Restriction** This option is valid only in OVERLAY and OVERLAYEQUATED layouts.
This option overrides the AXISLINEEXTENT= option in the BEGINGRAPH statement.

Tip
The graph wall outline might appear to be an axis line. In that case, use the WALLDISPLAY=NONE or WALLDISPLAY=(FILL) option in the layout statement to suppress the wall outline.

**LOGOPTS=(log-axis-options)**
specifies one or more options for a log axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction This option is ignored if the axis type is not LOG.

See “Options for Log Axes Only” on page 1028 for the options that you can use for log-axis-options.

**NAME="string"**
assigns a name to an axis for reference in other statements. Currently, it is used only in an AXISLEGEND statement.

Restriction This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interactions This option is ignored unless the axis is discrete. The axis can be discrete by default, or explicitly set to discrete with a TYPE= DISCRETE setting.

For this option to take effect, an axis legend must be enabled. To enable an axis legend, the DISCRETEOPTS= option must set the TICKVALUEFITPOLICY to either EXTRACT or EXTRACTALWAYS. In addition, an AXISLEGEND statement must be specified to generate the axis legend.

**OFFSETMAX=AUTO | AUTOCOMPRESS | number**
reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.

AUTO
reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

number
specifies the offset as a decimal proportion of the full axis length.

Default AUTO

Range 0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

See “Adjusting Axis Offsets” on page 985

**OFFSETMIN=AUTO | AUTOCOMPRESS | number**
reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.
AUTO
reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

number
specifies the offset as a decimal proportion of the full axis length.

Default AUTO
Range 0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.
See “Adjusting Axis Offsets” on page 985.

REVERSE=TRUE | FALSE
specifies whether tick values should appear in the reverse order.

Default FALSE
See “boolean ” on page 1479 for other Boolean values that you can use.

SHORTLABEL="string"
specifies an alternate axis label to display when the default label or the label specified by the LABEL= option is too long to fit the available space. When LABELPOSITION=CENTER (default), the available space for an axis label is the full axis, including the axis offsets. When LABELPOSITION=DATA CENTER, the available space for an axis label is the axis tick display area, which excludes the axis offsets. If the label length exceeds the available space, then the label is anchored at the left or bottom offset. It extends beyond the opposing offset until it reaches the end of the axis where it is truncated. An ellipsis designates the truncation.

Interactions This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

This option has effect only when the LABELPOSITION= option is set to CENTER or DATACENTER.

Note If the specified label is itself too long for the axis, it is truncated in the display.

TICKSTYLE=OUTSIDE | INSIDE | ACROSS
specifies the placement of tick marks in relation to the axis line. The figure shows the tick display for each value.

![Tick Display Examples](image-url)
OUTSIDE
displays tick marks outside of the axis frame.

INSIDE
displays tick marks inside the axis frame.

ACROSS
displays tick marks across the axis line.

Default The GraphAxisLines:TickDisplay style reference.

Interaction This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick marks.

Notes This option has no effect on the placement of the tick values, which are always outside the axis frame.

This option applies to both major ticks and minor ticks.

TICKVALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the axis tick values.

Default The GraphValueText style element.

Interaction This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick values.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

TICKVALUEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.

<table>
<thead>
<tr>
<th>LEFT</th>
<th>CENTER</th>
<th>RIGHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notebooks</td>
<td>Notebooks</td>
<td>Notebooks</td>
</tr>
<tr>
<td>Ink</td>
<td>Ink</td>
<td>Ink</td>
</tr>
<tr>
<td>Printer paper</td>
<td>Printer paper</td>
<td>Printer paper</td>
</tr>
<tr>
<td>Staples</td>
<td>Staples</td>
<td>Staples</td>
</tr>
<tr>
<td>Pens</td>
<td>Pens</td>
<td>Pens</td>
</tr>
</tbody>
</table>

Defaults RIGHT for a Y axis

LEFT for a Y2 axis

Restriction This option is valid for the Y and Y2 axes only.

TICKVALUEVALIGN=TOP | CENTER | BOTTOM
specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.
Defaults

TOP for an X axis

BOTTOM for an X2 axis

Restriction

This option is valid for the X and X2 axes only.

**TIMEOPTS=(time-axis-options)**

specifies one or more options for a time axis.

Requirements

Columns associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a `name = value` pair and each pair is space separated.

Interaction

This option is ignored if the axis type is not TIME.

See

“Options for Time Axes Only” on page 1037 for the options that you can use for `time-axis-options`.

**TYPE=**AUTO | DISCRETE | LINEAR | TIME | LOG

specifies the type of axis to use.

**AUTO**

requests that the axis type be automatically determined, based on the overlay contents.

For more information, see “When Plots Share Data and a Common Axis” on page 979.

**DISCRETE**

uses a DISCRETE axis if possible.

The data for discrete axes can be character or numeric. You can add a `DISCRETEOPTS=` option list to customize this axis type.

**LINEAR**

uses a LINEAR axis if possible.

You can add a `LINEAROPTS=` option list to customize this axis type.

**TIME**

uses a TIME axis if possible.
Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a `TIMEOPTS=` option list to customize this axis type.

**LOG**

uses a LOG axis if possible.

You can add a `LOGOPTS=` option list to customize this axis type.

**Interaction**

If a log axis is requested and the axis data contains 0 or negative values, the axis reverts to a linear axis. This outcome can occur for the response axis of a bar chart, line chart, needle plot, or waterfall chart when a baseline intercept of 0 or less is specified. It can also occur for the response axis of a waterfall chart when an initial bar value of 0 or less is specified. To get a log response axis in those cases, set the baseline intercept or initial bar value to a positive value.

**Default**

`AUTO`

**Interactions**

If this option is set to anything other than `AUTO`, then plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 981.

After the axis type is determined (whether you set a specific type or `AUTO` is in effect), you can use only options that are supported by that axis type. For example, if `TYPE=TIME`, then only the general OVERLAY axis options and those available on `TIMEOPTS=` are supported.

### Options for Discrete Axes Only

The options that are documented in this section can be used with the `DISCRETEOPTS=` axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Discrete Axis Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>COLORBANDS</code></td>
<td>Specifies the display of alternating wall-color bands corresponding to the discrete axis bins.</td>
</tr>
<tr>
<td><code>COLORBANDSATTRS</code></td>
<td>Specifies the appearance of the alternating wall-color bands.</td>
</tr>
<tr>
<td><code>SORTORDER</code></td>
<td>Specifies how the values on the axis are to be sorted.</td>
</tr>
<tr>
<td><code>TICKDISPLAYLIST</code></td>
<td>Specifies the text that is displayed for the tick values that are defined in the <code>TICKVALUELIST=</code> option.</td>
</tr>
<tr>
<td><code>TICKTYPE</code></td>
<td>Specifies the position of the axis tick mark.</td>
</tr>
</tbody>
</table>
### Discrete Axis Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision on an axis.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the list of tick values that are displayed on the axis.</td>
</tr>
<tr>
<td>TICKVALUEROTATION</td>
<td>Specifies how the tick values are rotated on the X and X2 axes.</td>
</tr>
<tr>
<td>TICKVALUESPLITCHAR</td>
<td>Specifies whether the split characters are included in the displayed tick values.</td>
</tr>
<tr>
<td>TICKVALUESPLITCHARDROP</td>
<td>Specifies whether the split characters are included in or dropped from the displayed tick value.</td>
</tr>
<tr>
<td>TICKVALUESPLITJUSTIFY</td>
<td>Specifies justification of the strings that are inside the tick value block.</td>
</tr>
</tbody>
</table>

### COLORBANDS

COLORBANDS=NONE | EVEN | ODD

Specifies the display of alternating wall-color bands corresponding to the discrete axis bins.

- **Default**: NONE
- **Restriction**: This option applies to discrete axes only.
- **Interaction**: Specifying this option for more than one axis in the layout might have unexpected results. The order in which color bands are drawn might not match the order in which the axis options are specified.
- **Note**: The full width of a color band is the distance between midpoints. When no axis offsets are specified, the first band begins at one-half of the midpoint distance, and the last band ends at one-half of the midpoint distance. When axis offsets are specified, the first and last color bands on the axis might extend into their adjacent offsets by as much as half the color-band width.
- **Tips**: Borders for the color bands can be added by setting TICKTYPE=INBETWEEN in the DISCRETEOPTS= option, and by setting GRIDDISPLAY= ON. Because alternating color bands are drawn on top of the plot wall, this option can be coordinated with the LAYOUT OVERLAY statement’s WALLCOLOR= option.

### COLORBANDSATTRS

COLORBANDSATTRS=style-element | style-element (fill-options) | (fill-options)

Specifies the appearance of the alternating wall-color bands. For the alternating colors, one set uses the WALLCOLOR= colors that are set in the LAYOUT OVERLAY statement, and the other set uses the colors set on this option.
The GraphBlock style element.

This option applies to discrete axes only.

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

**SORTORDER=**

specifies how the axis values are to be sorted.

**Note:** This option applies to SAS 9.4M5 and to later releases.

**AUTO**

for a column in a SAS data set, enables the system to determine the best sorting for the axis tick values.

For a column in a CAS in-memory table, sorts the tick values in ascending order by the formatted values.

**DATA**

arranges the axis tick values in the order in which they appear in the data.

**Note** In SAS 9.4M6 and SAS 9.4M5, when data stored in a CAS in-memory table is used and DATA is specified, unpredictable results might occur. To generate consistent graphs in that case, specify AUTO. Starting with SAS Viya 3.5 and SAS Studio 5.2, when data stored in a CAS in-memory table is used and DATA is specified, DATA is ignored, and AUTO is used instead.

**Default** AUTO

**TICKDISPLAYLIST=(string-list)**

specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the TICKVALUELIST= option. The strings map one-to-one positionally with the values that are listed in the TICKVALUELIST= option.

**Default** Determined by the system or by the TICKVALUELIST= option.

**Requirements**

The list of values must be enclosed in parentheses.

Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.

**Tip**

This option should be used with the TICKVALUELIST= option. The number of items in the list for this option should equal the number of items in the list for the TICKVALUELIST= option.

**Example**

The following example specifies the axis tick values 10, 20, 30, and 40, and the tick display values A, B, C, and D:

```
tickvaluelist=('10' '20' '30' '40');
tickdisplaylist=('A' 'B' 'C' 'D');
```
TICKTYPE=MIDPOINT | INBETWEEN
specifies the position of the axis tick marks.

MIDPOINT
places the tick marks at the midpoint value location.

INBETWEEN
places the tick marks half way between adjacent midpoint locations.

Default    MIDPOINT
Restriction This option applies to discrete axes only.
Note       Starting with SAS 9.4M2, when TICKTYPE=INBETWEEN, the outermost tick marks and grid lines at each end of the axis are not drawn.

TICKVALUEFITPOLICY=policy
specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

EXTRACT
displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see "Extracting Discrete Axis Tick Values into a Legend" in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see "Extracting Discrete Axis Tick Values into a Legend" in SAS Graph Template Language: User’s Guide.

NONE
makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

SPLIT
splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See  TICKVALUESPLITCHAR=
SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit the available space.

THIN
eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

EXTRACT
display consecutive integers along the axis instead of the actual tick values to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.
ROTATETHIN
attempts the ROTATE policy, and then the THIN policy.

SPLIT
splits the tick value at a split character, which is specified by the
TICKVALUESPLITCHAR= option, only when necessary in order to make the
value fit the available space.
A split does not occur at a split character if a split is not needed at that
location. If the value does not contain any of the specified split characters,
then the value is not split. Values that are not split or that do not fit the
available space even after splitting might overlap the adjoining space.

See TICKVALUESPLITCHAR=

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is
specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITROTATE
attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

Note: This feature applies to SAS 9.4M5 and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.
Thinning is performed when the tick values do not fit the available space.

Note: This feature applies to SAS 9.4M5 and to later releases.

STAGGER
alters the tick values between two rows.

STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.
TRUNCATETHIN

attempts the TRUNCATE policy, and then the THIN policy.

Defaults

<table>
<thead>
<tr>
<th>Policy</th>
<th>Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROTATE</td>
<td>X and X2 axes</td>
</tr>
<tr>
<td>THIN</td>
<td>Y and Y2 axes</td>
</tr>
</tbody>
</table>

Note

A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=\texttt{format}

specifies how to format the values for major tick marks.

Note: This feature applies to \texttt{SAS 9.4M3} and to later releases.

Restrictions

This option applies only to discrete axes.

Interaction

This option is ignored when the axis tick values are extracted to an axis legend. See \texttt{TICKVALUEFITPOLICY=EXTRACT | EXTRACTALWAYS} on page 1005.

Tip

Use this option when you want to duplicate tick values on an axis.

TICKVALUELIST=\texttt{(string-list)}

specifies the list of tick values that are to be displayed on the axis.

\texttt{string-list}

a space-separated list of values, enclosed in parentheses.

You must enclose each value in the list in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

Requirements

The list of values must be enclosed in parentheses.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes

If the string list contains duplicate values, then the first occurrence of the duplicated value in the list is honored and the remaining instances are ignored.

When the values specified in the list are compared with the actual data values, leading blanks are honored and trailing blanks are ignored.

Tips

You can use this option to subset the axis values or to display the values in a specific order.

You can use this option to display values on the axis that are not contained in the data.
Examples

The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

tickvaluelist=("Sedan" "Sports" "Wagon" "SUV")

The following example specifies the axis tick values 10, 20, 30, and 40:

tickvaluelist=("10" "20" "30" "40")

TICKVALUEROTATION=DIAGONAL | DIAGONAL2 | VERTICAL

specifies how the tick values are rotated on the X and X2 axes.

DIAGONAL

rotates the tick values to a 45-degree diagonal position.

The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

DIAGONAL2

rotates the tick values to a –45-degree diagonal position.

The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

Note: This feature applies to SAS 9.4M5 and to later releases.

VERTICAL

rotates the labels to a 90-degree vertical position.

The labels are always drawn from bottom to top.

Default DIAGONAL

Restriction This option is valid for XAXISOPTS= and X2AXISOPTS= only.

Interaction The TICKVALUEFITPOLICY= option must be set to ROTATE or ROTATEALWAYS for this option to have any effect.

TICKVALUESPLITCHAR="character-list"

specifies a list of characters on which the tick values can be split, if needed. When multiple characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the tick value. In that case, all of the specified split characters together are treated as a single split character.

When TICKVALUESPLITPOLICY=SPLIT, if a tick value collision is detected, then the tick value is split on a split character only if necessary at that point in order to avoid collision. In that case, a split might not occur on every split character. When TICKVALUEFITPOLICY=SPLITALWAYS, the tick value is split unconditionally on every occurrence of a split character. If the tick value does not contain any of the specified split characters, then it is not split.

“character-list”

one or more characters with no delimiter between each character.

Default A blank space

Requirements The list of characters must be enclosed in quotation marks.
Multiple characters must be specified with no delimiters. For example, to specify the characters a, b, and c, use the following option:

```
tickvaluesplitchar="abc"
```

**Interactions**
This option is ignored unless option `TICKVALUEFITPOLICY=` is set to SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.

The `TICKVALUEFITPOLICY=` option sets the policy that is used to manage the split behavior of the tick values.

The `TICKVALUESPLITCHARDROP=` option specifies whether the split characters are displayed or dropped from the display.

**Notes**
When multiple characters are specified, the order of characters in the list is not significant.

The split characters are case sensitive.

**Tips**
Use the `TICKVALUESPLITCHARDROP=` option to specify the justification of the strings in the tick value block.

For the X and X2 axis tick values, use the `TICKVALUEVALIGN=` option to specify the vertical alignment of the tick values.

For the Y and Y2 axis tick values, use the `TICKVALUEHALIGN=` option to specify the horizontal alignment of the tick values.

**Example**
The following example specifies a blank space, a comma, and an underscore as split characters:

```
tickvaluesplitchar=" , _"
```

**TICKVALUESPLITCHARDROP=TRUE | FALSE**
specifies whether the split characters should be included in the displayed tick values. The split characters are specified by the `TICKVALUESPLITCHAR=` option.

**TRUE**
drops the split characters from the tick value display.

The following figure shows an example in which `TICKVALUESPLITCHARDROP=TRUE` and three-word, asterisk-delimited tick values are split on the asterisk character by using the SPLITALWAYS policy.

```
| Product Group A | Product Group B | Product Group C | Product Group D | Product Group E |
```

Notice that the asterisk delimiter is not displayed.

**FALSE**
includes the split characters in the tick value display.

The fit policy determines how the characters are displayed. If the display policy is SPLIT or SPLITTHIN and `TICKVALUESPLITCHARDROP=FALSE`, then each tick value is split at a split character only where a split is necessary in order to make the value fit the available space. A split might not occur at
every split character in the tick value. At each split point, the split character
remains as the last character in the current line. The characters that follow
the split character, up to and including the split character at the next split
point, are then wrapped to the following line. This process repeats until the
entire data tick value is displayed. The following figure shows an example in
which TICKVALUESPLITCHARDROP=FALSE and three-word, asterisk-
delimited tick values are split on the asterisk character by using the SPLIT
policy.

Notice that a split occurs on the first asterisk and not at the second. In this
case, a split is not needed at the second asterisk.

If the fit policy is SPLITALWAYS or SPLITALWAYSTHIN, and
TICKVALUESPLITCHARDROP=FALSE, then each tick value is split at every
instance of a split character in the value regardless of whether a split is
actually needed. Each split character remains as the last character in the
current line. The characters that follow each split character, up to and
including the next split character, are then wrapped to the next line. The
following figure shows an example in which
TICKVALUESPLITCHARDROP=FALSE and three-word, asterisk-delimited
tick values are split on the asterisk character by using the SPLITALWAYS
policy.

Notice that a split occurs after each asterisk and each asterisk appears at the
end of the line. In this case, three lines are displayed.

Default TRUE

Interactions The TICKVALUESPLITCHAR= option specifies the split character
or characters.

This option is ignored unless option TICKVALUEFITPOLICY= is set to SPLIT, SPLITALWAYS, SPLITALWAYSTHIN, or SPLITALWAYSTHIN.

See “boolean “ on page 1479 for other Boolean values that you can use.

TICKVALUESPLITJUSTIFY= CENTER | LEFT | RIGHT

specifies justification of the strings that are inside the tick value block. The
justification is relative to an individual tick value’s display area and does not
affect the display of tick values that are not split.
Defaults

- CENTER for an X or X2 axis
- RIGHT for a Y axis
- LEFT for a Y2 axis

Interaction

This option is ignored unless option `TICKVALUEFITPOLICY= SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.`

---

### Options for Linear Axes Only

The options that are documented in this section can be used only with the `LINEAROPTS=` axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Linear Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDERANGES on page 1013</td>
<td>Specifies one or more ranges for a broken axis.</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Specifies that evenly spaced integer values are used for tick marks.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKCOUNT</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether the minor tick marks are displayed on the axis.</td>
</tr>
<tr>
<td>ORIGIN</td>
<td>Specifies that the axis perpendicular to the current axis be drawn at the indicated data value.</td>
</tr>
<tr>
<td>THRESHOLDMAX</td>
<td>Specifies a bias for including one more tick mark at the maximum end of the axis.</td>
</tr>
</tbody>
</table>
## Linear Axis Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>THRESHOLDMIN</td>
<td>Specifies a bias for including one more tick mark at the minimum end of the axis.</td>
</tr>
<tr>
<td>TICKDISPLAYLIST</td>
<td>Specifies the text that is displayed for the tick values that are defined by the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision. Only the default policy (THIN) is available for a Y or Y2 axis.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values for a linear axis as list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification can extend the axis data range.</td>
</tr>
<tr>
<td>TICKVALUESEQUENCE</td>
<td>Specifies the tick values for a linear axis by start, end, and increment.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display.</td>
</tr>
</tbody>
</table>

**INCLUDERANGES=(start–end <start2–end2 startN–endN ...>)**

specifies the ranges for a broken axis.

---

**Note:** This option is valid starting in **SAS 9.4M1**.

**start**

- specifies a numeric value or the keyword MIN.
  - MIN specifies the minimum data value.

**end**

- specifies a numeric value or the keyword MAX.
  - MAX specifies the maximum data value.

The following figure shows a linear axis, broken into ranges 50–52 and 56–73.
The following figure shows a time axis, broken into ranges '01Jan2001'd–'01May2003'd and '01Jan2005'd–'01Oct2005'd.

As shown in the figure, break lines are drawn to indicate the break in the axis.

**Restrictions**

This option is valid for linear and time axes in an OVERLAY layout only.

Only one axis can be broken. If this option is specified for both axes, then it is honored for the vertical axis and ignored for the horizontal axis.

When plots are associated with the X and X2 axes or with Y and Y2 axes, neither axis can be broken.

A binned heat map or histogram axis cannot be broken.

A broken axis is not supported in vector graphics output. When a broken axis is specified and vector graphics output is requested, the graph is converted into an image instead. A note indicating the conversion is written to the SAS log. To restore the vector graphics output in that case, remove the INCLUDERANGES= option from the LINEAROPTS= or TIMEOPTS= option.

**Requirements**

All of the ranges must be enclosed in parenthesis.

You must specify each range as a starting value, a hyphen, and an ending value. You must separate adjacent ranges with a space.

Each range must be nonzero. A zero range such as 12–12 is considered invalid.

**Interactions**

When this option is specified, axis options THRESHOLDMIN=, THRESHOLDMAX=, VIEWMIN=, VIEWMAX=, and TICKVALUEPRIORITY= are ignored. Suboption EXTRACTSCALE= of the TICKVALUEFORMAT= option is also ignored.

When this option is specified, the plot statement TIP= and URL= options are ignored.
Notes  When this option is specified, data-clipping might occur for the following: plot markers and marker characters, box-plot outlier markers, fixed-position data labels, needle plots and fringe plots in the X direction, reference lines and drop lines on the broken axis, axis tables, and relative bubble plots.

Curve label positions are based on the contiguous axis data range. When curve labels are specified with a broken axis, the curve label positions might not be ideal.

Tip  Starting with SAS 9.4M3, you can use the AXISBREAKTYPE= and AXISBREAKSYMBOL= options in the BEGINGRAPH statement to display the break in the axis as only a symbol on the axis line.

See  “Creating a Broken Linear Axis” in SAS Graph Template Language: User’s Guide

Example  includeranges=(10-500 1000-5000 10000-50000)

**INTEGER=TRUE | FALSE**
specifies that evenly spaced integer values are used for tick marks.

Default  FALSE

Interactions  This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option.

INTEGER=TRUE is ignored for the X or X2 axis when a histogram plot is the primary plot and BINAXIS=TRUE is specified in the HISTOGRAM or HISTOGRAMPARM statement.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**MINORGRID=TRUE | FALSE**
specifies whether grid lines are displayed at the minor tick marks.

![Diagram showing grid lines at minor tick marks for TRUE and FALSE]

Defaults  FALSE in SAS 9.4M1 and earlier releases.
The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

**Interaction**

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

**Tips**

The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**MINORGRIDATTRS=**

- `style-element` | `style-element (line-options)` | `(line-options)`

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

![Minor Grid Lines Example](image)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

**Defaults**

The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

**Interaction**

This option is ignored when MINORTICKS=FALSE.

**Note**

When `style-element` is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

**Tip**

Use the GRIDATTRS= option to control the appearance of the major grid lines.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

**Example**

Here is an example that specifies light blue, dotted lines for the minor grid.

```plaintext
minorgridattrs=(color=lightblue pattern=dot);
```
MINORTICKCOUNT=positive-integer
specifies the number of minor ticks that are displayed on the axis.

Defaults
Four ticks with five intervals in SAS 9.4M1 and earlier releases.
One tick with two intervals starting with SAS 9.4M2.

Interactions
The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

Tip
To display \( n \) intervals between major ticks, use MINORTICKCOUNT=\( n-1 \).

MINORTICKS=TRUE | FALSE
specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

TRUE
\[ \begin{array}{cccc}
0 & 50 & 100 & 150 & 200 \\
\end{array} \]
FALSE
\[ \begin{array}{cccc}
0 & 50 & 100 & 150 & 200 \\
\end{array} \]

Default
FALSE

Interaction
If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Tip
Use the MINORGRID= option to display grid lines at the minor tick values.

See
“boolean ” on page 1479 for other Boolean values that you can use.

ORIGIN=number
specifies that the axis perpendicular to the current axis is drawn at the indicated data value. For managing origin settings, GTL treats the X and Y axes as a pair, and the X2 and Y2 axes as a separate pair. Thus, if you set the Y-axis origin to 200, then the X axis is drawn from that origin point. If the graph also displays an X2 axis, then it is unaffected and does not move. Similarly, if you set an origin for the Y2 axis, then the X2 axis moves to that origin point and the X axis is unaffected.

If you set an origin for the Y2 axis and there is no X2 axis, then the origin setting for Y2 does not affect the graph display. That is, the X axis does not move to that origin point.

If you set an origin for an axis and the axis has a tick value at that origin value, the tick value is not displayed. Suppressing the tick value at the origin prevents the value from colliding with the axis value on the perpendicular axis. However, it is possible that the tick values on the orthogonal axes will collide.
The axis perpendicular to the current axis is drawn at the minimum tick value minus the OFFSETMIN= value.

Restriction
This option applies to linear axes only.

Interactions
If the specified value is outside the data range for the current axis, then the data range is extended to include the value.

The axis line, ticks, and tick values of the "perpendicular" axis move to the location indicated by the origin. The axis label is not moved.

Tip
This option is often used to create Cartesian axes (axes centered at ORIGIN=0).

**THRESHOLDMAX=number**
specifies a bias for including one more tick mark at the maximum end of the axis.

| Default | 0.30 |
| Range   | 0–1  |

Restriction
This option applies to linear axes only.

Interactions
This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

This option is ignored when the INCLUDERANGES= option is specified.

Tips
If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0.

See
"Adjusting Axis Thresholds" on page 984

**THRESHOLDMIN=number**
specifies a bias for including one more tick mark at the minimum end of the axis.

| Default | 0.30 |
| Range   | 0–1  |

Restriction
This option applies to linear axes only.

Interactions
This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

This option is ignored when the INCLUDERANGES= option is specified.
Tips

If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0.

See “Adjusting Axis Thresholds” on page 984

TICKDISPLAYLIST=(string-list)

specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the TICKVALUELIST= option. The strings map one-to-one positionally with the values that are listed in the TICKVALUELIST= option.

Default

The display of tick values is controlled by the TICKVALUEFORMAT= option.

Requirements

The list of values must be enclosed in parentheses.

Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.

Interaction

When this option is specified, the TICKVALUEFORMAT= option is ignored.

Tip

This option should be used with the TICKVALUELIST= option. The number of items in the list for this option should equal the number of items in the list for the TICKVALUELIST= option.

TICKVALUEFITPOLICY=policy

specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

EXTRACT

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement

The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS

same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.
The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

### NONE

makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

### SPLIT

splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See \[
\text{TICKVALUESPLITCHAR=}
\]

### SPLITALWAYS

always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See \[
\text{TICKVALUESPLITCHAR=}
\]

### SPLITALWAYSTHIN

same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

### SPLITTHIN

same as SPLIT, except that thinning is performed when long words do not fit the available space.

### THIN

eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

### EXTRACT

display consecutive integers along the axis instead of the actual tick values to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

#### Requirement

The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

### EXTRACTALWAYS

same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

#### Requirement

The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting
NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The **TICKVALUEROTATION=** option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The **TICKVALUEROTATION=** option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.

ROTATETHIN
attempts the ROTATE policy, and then the THIN policy.

SPLIT
splits the tick value at a split character, which is specified by the **TICKVALUESPLITCHAR=** option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See **TICKVALUESPLITCHAR=**

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the **TICKVALUESPLITCHAR=** option.

See **TICKVALUESPLITCHAR=**

SPLITROTATE
attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

---

**Note:** This feature applies to **SAS 9.4M5** and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

---

**Note:** This feature applies to **SAS 9.4M5** and to later releases.
STAGGER
alternates the tick values between two rows.

STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.

TRUNCATETHIN
attempts the TRUNCATE policy, and then the THIN policy.

Default     THIN

Note       A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=(format-options) | DATA | format
specifies how to format the values for major tick marks.

(format-options)
specifies one or more formatting options for major tick values.
Together, these options provide parameters for determining an optimal format
(w.d, Ew., BESTw.) for displaying major tick values.

MAXWIDTH=integer
specifies the maximum width for displayed tick values.
Values might be rounded or converted to E notation to fit into this width.

Default  8

MAXDECIMALS=integer
specifies the maximum number of decimals for displayed tick values.
Values might be rounded or converted to E notation to fit into this width.

Default  6

Note       The MAXWIDTH= option value should be greater than the MAXDECIMALS= option value.

PREFERREDDECIMALS=integer
specifies the number of decimal places that you want to display for most values.
The actual number might vary based on other constraints.

Default  2
**EXTRACTSCALE**=TRUE | FALSE
specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.

The scale can be a named scale or a scientific-notation scale. The **EXTRACTSCALETYPE**= option specifies the scale type. The scale that is used is appended to the axis label, as shown in the following example.

**Total Sales (millions)**

For long axis labels, if the scale does not fit the available space, then the label is truncated, and the scale is appended to the truncated label. Ellipses indicate that the label was truncated, as shown in the following example.

**Total Sales for the Fourth Quarter Of ... (millions)**

In extreme cases in which the scale does not fit even with truncation, the entire axis is dropped.

**Default** FALSE

**Restriction** The scale that is extracted by the **EXTRACTSCALE**= option is derived from the English locale for all locales.

**Interactions** The scale type is determined by the **EXTRACTSCALETYPE**= option.

If the axis label is not displayed, then the **EXTRACTSCALE**=TRUE option is ignored.

The **EXTRACTSCALE**= suboption is ignored when the **INCLUDERANGES**= option is specified.

**Note** When **EXTRACTSCALE**=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

**See** "boolean " on page 1479 for other Boolean values that you can use.

**EXTRACTSCALETYPE**=DEFAULT | SCIENTIFIC
specifies whether to extract a named scale or a scientific-notation scale.

**DEFAULT** extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as \(10^n\)) for values over 999 trillion. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of \(1/10\) (\(10^{-n}\)) for values less than 1 trillionth.

**SCIENTIFIC** extracts a scientific-notation scale.
A scientific-notation scale is a multiple of 10 expressed as $10^n$ for values greater than 1, or a multiple of $1/10$ expressed as $10^{-n}$ for values less than 1.

**Default**  
DEFAULT

**Restriction**  
The scale is derived from the English locale for all locales.

**DATA**  
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

**format**  
specifies a format to apply to the major tick values.

**Restriction**  
GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

**Note**  
If you specify a format that significantly reduces precision, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

**Default**  
(MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE, EXTRACTSCALETYPE=DEFAULT)

**Interaction**  
This option is ignored when the TICKDISPLAYLIST= option is specified.

**TICKVALUELIST=(numeric-list)**  
specifies the tick values for a linear axis as a list.

**Default**  
An internal algorithm determines the tick marks, based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default, when this option is used, the only tick values that appear are the tick values in numeric-list that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or the implicit data range (set by the actual data minimum and data maximum).

**Restriction**  
This option applies to linear axes only.

**Requirement**  
The tick values must be specified as a space-separated list of numeric values, enclosed in parentheses.

**Interactions**  
This option overrides the INTEGER= option.

This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the DISPLAY= option or the DISPLAYSECONDARY= option does not display tick values.

The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, then all ticks in the tick list are displayed. This might result in some data not being displayed. For example, data might not be displayed
when the VIEWMIN= value is greater than the actual data minimum, or when the VIEWMAX= value is less than actual data maximum.

If TICKVALUEPRIORITY= TRUE, then the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by the numeric-list. The tick numeric-list can extend the implicit data range of the axis, but cannot reduce it.

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Tip

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

TICKVALUEPRIORITY=TRUE | FALSE

specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

TRUE

extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

FALSE

displays only the tick values that are specified by the TICKVALUELIST= option that fall within the explicit data range set by the VIEWMIN= and VIEWMAX= options or by the implicit data range set by the actual data minimum and data maximum.

Default

FALSE

Restriction

This option applies to linear axes only.

Interactions

When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored.

This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is not specified.

This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the tick values.

This option is ignored when the INCLUDERANGES= option is specified.

Note

If the minimum and maximum of the specified values are within the data range, then this option has no effect.

See

"boolean " on page 1479 for other Boolean values that you can use.

TICKVALUEROTATION=DIAGONAL | DIAGONAL2 | VERTICAL

specifies how the tick values are rotated on the X and X2 axes.
DIAGONAL
rotates the tick values to a 45-degree diagonal position.
The X labels read left to right in a downward direction. The X2 labels read left
to right in an upward direction.

DIAGONAL2
rotates the tick values to a –45-degree diagonal position.
The X labels read left to right in an upward direction. The X2 labels read left
to right in a downward direction.

Note: This feature applies to SAS 9.4M5 and to later releases.

VERTICAL
rotates the labels to a 90-degree vertical position.
The labels are always drawn from bottom to top.

Default: DIAGONAL

Restriction: This option is valid for XAXISOPTS= and X2AXISOPTS= only.

Interaction: The TICKVALUEFITPOLICY= option must be set to ROTATE or
ROTATEALWAYS for this option to have any effect.

TICKVALUESEQUENCE=(START=number END=number INCREMENT=number)
specifies the tick values by start, end, and increment.

START=number
specifies the value for the first tick mark.

END=number
specifies the value for the last tick mark.

INCREMENT=number
specifies the increment for intermediate tick marks between the first and last
tick marks.

The END value always controls the last tick mark. The interval between the
last tick mark and the previous tick mark might not necessarily be the
INCREMENT value.

Default: An internal algorithm determines the tick marks, based on the actual
axis data range or the data range established by the VIEWMIN= and
VIEWMAX= options. By default, when this option is used, the
only tick values that appear are those that fall within the explicit data
range (set by VIEWMIN= and VIEWMAX=) or the implicit data
range (set by the actual data minimum and data maximum).

Interaction: This option overrides the INTEGER= option.

The VIEWMIN= and VIEWMAX= options alter the axis data range. If
the VIEWMIN= option is set to the START= option value and the
VIEWMAX= option is set to the END= option value, then all ticks in
the tick sequence are displayed.

If TICKVALUEPRIORITY= TRUE, then the tick sequence might
extend the explicit data range of the axis, but never reduce it.
This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display tick marks.

Tip
The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

See
TICKVALUELIST= option as an alternative for customizing tick marks.

**VIEWMAX=number**
specifies the maximum data value to include in the display. The value might be adjusted by the threshold calculation.

Default
The maximum value in the data for the specified axis.

Interactions
This option does not determine the maximum axis tick value that is displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

This option is ignored when TICKVALUEPRIORITY= TRUE.

This option is ignored when the INCLUDERANGES= option is specified.

Notes
Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip
To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

See
"Adjusting the Axis View" on page 983

**VIEWMIN=number**
specifies the minimum data value to include in the display. The value might be adjusted by the threshold calculation.

Default
The minimum value in the data for the specified axis.

Interactions
This option does not determine the minimum axis tick value that is displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

This option is ignored when TICKVALUEPRIORITY= TRUE.

This option is ignored when the INCLUDERANGES= option is specified.

Notes
Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such
as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

**Tip**
To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

**See**
“Adjusting the Axis View” on page 983

### Options for Log Axes Only

The options that are documented in this section can be used with the LOGOPTS= axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Log Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE</td>
<td>Specifies the base of the logarithmic scale for the axis values.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKCOUNT</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>THRESHOLDMAX</td>
<td>Specifies a bias for including one more tick mark at the maximum end of the axis.</td>
</tr>
<tr>
<td>THRESHOLDMIN</td>
<td>Specifies a bias for including one more tick mark at the minimum end of the axis.</td>
</tr>
<tr>
<td>TICKINTERVALSTYLE</td>
<td>Specifies how to scale and format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT=</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the tick values for a log axis as a space-separated list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether the TICKVALUELIST= specification can extend the axis data range.</td>
</tr>
<tr>
<td>VALUESTYPE</td>
<td>Specifies the scale that the system uses when interpreting the TICKVALUELIST=, VIEWMAX=, and VIEWMIN= option values.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
</tbody>
</table>
Log Axis Option | Description
--- | ---
**VIEWMIN** | Specifies the minimum data value to include in the display.

**BASE=** 10 | 2 | E
specifies the base of the logarithmic scale for the axis values.

Default

Restriction This option applies to log axes only.

**MINORGRID=** TRUE | FALSE
specifies whether grid lines are displayed at the minor tick marks.

Defaults FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See “boolean” on page 1479 for other Boolean values that you can use.

**MINORGRIDATTRS=** style-element | style-element (line-options) | (line-options)
specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)
The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

**Defaults**

The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

**Interaction**

This option is ignored when MINORTICKS=FALSE.

**Note**

When *style-element* is specified, only the style element's CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

**Tip**

Use the GRIDATTRS= option to control the appearance of the major grid lines.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

**Example**

Here is an example that specifies light blue, dotted lines for the minor grid.

```
minorgridattrs=(color=lightblue pattern=dot);
```

**MINORTICKCOUNT=positive-integer**

specifies the number of minor ticks that are displayed on the axis.

**Default**

Eight ticks with nine intervals (BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.).

**Restriction**

Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

**Interactions**

The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

**Tip**

To display *n* intervals between major ticks, use MINORTICKCOUNT=*n*-1.

**MINORTICKS=TRUE | FALSE**

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.
Default ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

Interaction

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform log tick values such as 10, 20, 30, and 40, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform log tick values such as 1, 10, 100, and 1000 in the TICKVALUELIST= option.

Tip

Use the MINORGRID= option to display grid lines at the minor tick values.

See

“boolean ” on page 1479 for other Boolean values that you can use.

**THRESHOLDMAX=** number

specifies a bias for including one more tick mark at the maximum end of the axis.

Default 0.30

Range 0–1

Restriction This option applies to log axes only.

Tips

If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

See

“Adjusting Axis Thresholds” on page 984

**THRESHOLDMIN=** number

specifies a bias for including one more tick mark at the minimum end of the axis.

Default 0.30

Range 0–1

Restriction This option applies to log axes only.

Tips

If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.
Specifying \texttt{THRESHOLDMIN=}1 and \texttt{THRESHOLDMAX=}1 ensures that the data range is bounded by tick marks.

See “Adjusting Axis Thresholds” on page 984

\textbf{TICKINTERVALSTYLE=}AUTO | LOGEXPAND | LOGEXPONENT | LINEAR

specifies how to scale and format the values for major tick marks.

\textbf{AUTO}

selects a LOGEXPAND, LOGEXPONENT, or LINEAR representation automatically based on the range of the data.

When the data range is small (within an order of magnitude), a LINEAR representation is typically used. Data ranges that encompass several orders of magnitude typically use the LOGEXPAND or LOGEXPONENT representation.

\textbf{LOGEXPAND}

places the major tick marks at uniform intervals at integer powers of the base.

The tick values are expanded as follows:

\begin{tabular}{l}
\textbf{Base=10} \\
\hline
1 & 10 & 100 & 1000 \\
\end{tabular}

\begin{tabular}{l}
\textbf{Base=2} \\
\hline
1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 & 512 & 1024 \\
\end{tabular}

\begin{tabular}{l}
\textbf{Base=E} \\
\hline
\text{e^0} & \text{e^1} & \text{e^2} & \text{e^3} & \text{e^4} & \text{e^5} & \text{e^6} & \text{e^7} \\
\end{tabular}

\textbf{LOGEXPONENT}

places the major tick marks at uniform intervals at integer powers of the base.

The tick values are only the integer exponents for all bases.

\begin{tabular}{l}
\hline
0 & 1 & 2 & 3 \\
\end{tabular}

\textbf{LINEAR}

places the major tick marks at non-uniform intervals that cover the range of the data.

\begin{tabular}{l}
\hline
10 & 40 & 200 & 400 & 600 & 1200 \\
\end{tabular}

\textbf{Default} AUTO

\textbf{Restrictions} This option applies to log axes only.

For LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LOGEXPAND, formats on data columns contributing to the axis are ignored, although any “named format” on the column is retained. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of \texttt{w.d}, \texttt{Ew}, or \texttt{BESTw}. Other formats (SAS defined or user-defined) are used if specified.
GTL currently honors most but not every SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

**Note**

When BASE=10 and LOGEXPAND or LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

**Tip**

If you use TICKINTERVALSTYLE=LOGEXPONENT, then you might want to include information in the axis label about which base is used.

**TICKVALUEFORMAT=DATA | format**

Specifies how to format the values for major tick marks.

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**DATA**

Uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

**format**

Specifies a format to apply to the major tick values.

**Restriction**

GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

**Restriction**

This option applies to log axes only.

**Interactions**

This option is ignored when TICKINTERVALSTYLE=LOGEXPONENT.

When TICKINTERVALSTYLE=LOGEXPAND, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When TICKINTERVALSTYLE=LINEAR, this option is honored for the base 10, base 2, and base E logarithmic scales.

**See**

BASE=

TICKINTERVALSTYLE=

**TICKVALUELIST=**\(\text{numeric-list}\)**

Specifies the tick values for a linear axis as a list.

**Default**

Only the tick values specified in the list that fall within the explicit data range set by the VIEWMIN= and VIEWMAX= options or by the implicit data range set by the actual data minimum and data maximum are displayed. An internal algorithm determines the tick marks.

**Requirements**

The tick values must be specified as a space-separated list of numeric values, enclosed in parentheses.
The values that you specify must be appropriate for the VALUESTYPE= specification. Otherwise, unexpected results might occur. If VALUESTYPE=EXPANDED is in effect (default), specify increments of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If VALUESTYPE=EXPONENT is in effect, specify integer increments of the log base power exponent such as 1, 2, 3, and so on.

Interactions

The VALUESTYPE= option determines how the values in the list are interpreted.

The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, then all ticks in the tick list are displayed. This might result in some data not being displayed. For example, data might not be displayed when the VIEWMIN= value is greater than the actual data minimum, or when the VIEWMAX= value is less than actual data maximum.

If the VIEWMIN= value is greater than the actual data minimum or the VIEWMAX= value is less than actual data maximum, some data might not be displayed.

This option is ignored if the DISPLAY= or the DISPLAYSECONDARY= option does not display the tick values.

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform log tick values such as 10, 20, 30, and 40, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform log tick values such as 1, 10, 100, and 1000 in the TICKVALUELIST= option.

See

VIEWMIN= and VIEWMAX= options for controlling the data range

TICKINTERVALSTYLE= for specifying the scale and format of the major tick values

TICKVALUEPRIORITY= for controlling the behavior of the TICKVALUELIST= option

BASE= for specifying the log base

**TICKVALUEPRIORITY=TRUE | FALSE**

specifies whether the TICKVALUELIST= specification can extend the axis data range.

**TRUE**

extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by the TICKVALUELIST= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

**FALSE**

displays only the tick values that are specified by the TICKVALUELIST= option that fall within the explicit data range set by the VIEWMIN= and
VIEWMAX= options or by the implicit data range set by the actual data minimum and data maximum.

**Default**
FALSE

**Interactions**
When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored.

This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the tick values.

This option is ignored if the TICKVALUELIST= option is not specified.

**Note**
If the minimum and maximum of the specified values are within the data range, then this option has no effect.

**See**
“boolean” on page 1479 for other Boolean values that you can use.

**VALUESTYPE=EXPANDED | EXPONENT**
specifies the scale that the system uses when interpreting the TICKVALUELIST=, VIEWMAX=, and VIEWMIN= option values. Use this option to choose your preferred way of specifying log-axis values.

**EXPANDED**
values are interpreted as powers of the base such as 0.1, 1, 10, 100, and so on, for base 10, for example.

**EXPONENT**
values are interpreted as integer exponents of the base such as 1, 2, 3, and so on, for base 10, base 2, and base E.

**Default**
EXPANDED

**Note**
This option does not change the style of the tick values that are displayed on the axis. It changes only how the VIEWMIN=, VIEWMAX=, and TICKVALUELIST= option values are interpreted by the system.

**Tip**
This option is particularly useful when BASE=E.

**Examples**
The following example specifies VIEWMIN= and VIEWMAX= as exponent values instead of as expanded values on an expanded Base 10 log axis. This results in X-axis tick values of 10, 100, 1000, 10000, and 100000.

```plaintext
xaxisopts=(type=log
    logopts=(base=10
        tickintervalstyle=logexpand
        valuestype=exponent
        viewmin=1 viewmax=5));
```

The following example specifies TICKVALUELIST= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```plaintext
xaxisopts=(type=log
    logopts=(base=10
        tickintervalstyle=logexponent
        tickvaluepriority=true
        valuestype=expanded
```
**VIEWMAX=number**

specifies the maximum data value to include in the display.

<table>
<thead>
<tr>
<th>Default</th>
<th>The maximum value in the data for the specified axis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement</td>
<td>The value that you specify must be appropriate for the VALUESTYPE= specification and the log base. Otherwise, unexpected results might occur. If VALUESTYPE=EXPANDED is in effect (default), specify an increment of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If VALUESTYPE=EXPONENT is in effect, specify an integer increment of the log base power exponent such as 1, 2, 3, and so on.</td>
</tr>
<tr>
<td>Interactions</td>
<td>This option is ignored when TICKVALUEPRIORITY= TRUE.</td>
</tr>
<tr>
<td>Notes</td>
<td>Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.</td>
</tr>
<tr>
<td></td>
<td>If an invalid value is specified for the VIEWMAX= option, the default value for VIEWMAX= is used instead. In that case, if the default value for VIEWMAX= is less than the value specified by the VIEWMIN= option, then the VIEWMIN= and VIEWMAX= values are swapped.</td>
</tr>
<tr>
<td></td>
<td>The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.</td>
</tr>
<tr>
<td></td>
<td>When BASE=10 and TICKINTERVALSTYLE=LOGEXPAND or TICKINTERVALSTYLE=LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.</td>
</tr>
<tr>
<td>Tip</td>
<td>To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>See</td>
<td>&quot;Adjusting the Axis View&quot; on page 983</td>
</tr>
<tr>
<td>Examples</td>
<td>The following example specifies a value of 100,000 as an expanded value on a base 10 log axis: VIEWMAX=100000</td>
</tr>
<tr>
<td></td>
<td>The following example specifies a value of 100,000 as an exponent value on a base 10 log axis: VIEWMAX=5</td>
</tr>
</tbody>
</table>

**VIEWMIN=number**

specifies the minimum data value to include in the display.

| Default                      | The minimum value in the data for the specified axis. |
Requirement

The value that you specify must be appropriate for the `VALUESTYPE=` specification and the log base. Otherwise, unexpected results might occur. If `VALUESTYPE=EXPANDED` is in effect (default), specify an increment of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If `VALUESTYPE=EXPONENT` is in effect, specify an integer increment of the log base power exponent such as 1, 2, 3, and so on.

Interactions

This option is ignored when `TICKVALUEPRIORITY=TRUE`.

Notes

Setting a `VIEWMAX=` or `VIEWMIN=` value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the `VIEWMIN=` value. The `VIEWMIN=` and `VIEWMAX=` values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

When `BASE=10` and `TICKINTERVALSTYLE=LOGEXPAND` or `TICKINTERVALSTYLE=LOGEXPONENT` is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

Tip

To display the `VIEWMIN=` value as the minimum tick value, use the `TICKVALUELIST=` option.

See

“Adjusting the Axis View” on page 983

Examples

The following example specifies a value of 0.1 as an expanded value on a base 10 log axis:

```plaintext
VIEWMIN=0.1
```

The following example specifies a value of 0.1 as an exponent value on a base 10 log axis:

```plaintext
VIEWMIN=-1
```

Options for Time Axes Only

The options that are documented in this section can be used with the `TIMEOPTS=` axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Time Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>INCLUDERANGES</code> on page 1038</td>
<td>Specifies one or more ranges for a broken axis.</td>
</tr>
<tr>
<td><code>INTERVAL</code></td>
<td>Specifies the time interval between major tick marks.</td>
</tr>
<tr>
<td><code>INTERVALMULTIPLIER</code></td>
<td>Specifies a multiplier to apply to the time interval that is in effect for the axis.</td>
</tr>
<tr>
<td>Time Axis Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKINTERVAL</td>
<td>Specifies the time interval between minor ticks.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>SPLITTICKVALUE</td>
<td>Specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available on a Y or Y2 axis.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available on a Y or Y2 axis.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUENAME</td>
<td>Specifies the order of the tick values for a time axis as list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification can extend the axis data range.</td>
</tr>
<tr>
<td>TICKVALUERotation</td>
<td>Specifies how the tick values are rotated on the X and X2 axes.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display.</td>
</tr>
</tbody>
</table>

**INCLUDERANGES=(start–end <start2–end2 startN–endN ...>)**
specifies the ranges for a broken axis.

Note: This option is valid starting in SAS 9.4M1.

- **start** specifies a SAS time, date, or date-time constant, or the keyword MIN. MIN specifies the minimum data value.
- **end** specifies a SAS time, date, or date-time constant, or the keyword MAX. MAX specifies the maximum data value.

The following figure shows a linear axis, broken into ranges 50–52 and 56–73.
The following figure shows a time axis, broken into ranges '01Jan2001'd–'01May2003'd and '01Jan2005'd–'01Oct2005'd.

As shown in the figure, break lines are drawn to indicate the break in the axis.

**Restrictions**

This option is valid for linear and time axes in an OVERLAY layout only.

Only one axis can be broken. If this option is specified for both axes, then it is honored for the vertical axis and ignored for the horizontal axis.

When plots are associated with the X and X2 axes or with Y and Y2 axes, neither axis can be broken.

A binned heat map or histogram axis cannot be broken.

A broken axis is not supported in vector graphics output. When a broken axis is specified and vector graphics output is requested, the graph is converted into an image instead. A note indicating the conversion is written to the SAS log. To restore the vector graphics output in that case, remove the INCLUDERANGES= option from the LINEAROPTS= or TIMEOPTS= option.

**Requirements**

All of the ranges must be enclosed in parenthesis.

You must specify each range as a starting value, a hyphen, and an ending value. You must separate adjacent ranges with a space.

Each range must be nonzero. A zero range such as 12–12 is considered invalid.

**Interactions**

When this option is specified, axis options THRESHOLDDMIN=, THRESHOLDDMAX=, VIEWMIN=, VIEWMAX=, and TICKVALUEPRIORITY= are ignored. Suboption EXTRACTSCALE= of the TICKVALUEFORMAT= option is also ignored.
When this option is specified, the plot statement TIP= and URL= options are ignored.

Notes

When this option is specified, data-clipping might occur for the following: plot markers and marker characters, box-plot outlier markers, fixed-position data labels, needle plots and fringe plots in the X direction, reference lines and drop lines on the broken axis, axis tables, and relative bubble plots.

Curve label positions are based on the contiguous axis data range. When curve labels are specified with a broken axis, the curve label positions might not be ideal.

Tip

Starting with SAS 9.4M3, you can use the AXISBREAKTYPE= and AXISBREAKSYMBOL= options in the BEGINGRAPH statement to display the break in the axis as only a symbol on the axis line.

See

"Creating a Broken Time Axis" in SAS Graph Template Language: User’s Guide

Example

```
INCLUDERANGES=('01Jan2001'd-'01May2003'd
 '01Jan2005'd-'01Oct2005'd)
```

**INTERVAL=**AUTO | SECOND | MINUTE | HOUR | DAY | TENDAY | WEEK | SEMIMONTH | MONTH | QUARTER | SEMIYEAR | YEAR

specifies the time interval between major ticks.

**Table 8.1 Time Intervals**

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>DATE, TIME, or DATETIME</td>
<td>automatically chosen</td>
<td>automatically chosen</td>
</tr>
<tr>
<td>SECOND</td>
<td>TIME or DATETIME</td>
<td>second</td>
<td>TIME8.</td>
</tr>
<tr>
<td>MINUTE</td>
<td>TIME or DATETIME</td>
<td>minute</td>
<td>TIME8.</td>
</tr>
<tr>
<td>HOUR</td>
<td>TIME or DATETIME</td>
<td>hour</td>
<td>TIME8.</td>
</tr>
<tr>
<td>DAY</td>
<td>DATE or DATETIME</td>
<td>day</td>
<td>DATE9.</td>
</tr>
<tr>
<td>TENDAY</td>
<td>DATE or DATETIME</td>
<td>10 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>WEEK</td>
<td>DATE or DATETIME</td>
<td>7 days</td>
<td>DATE9.</td>
</tr>
</tbody>
</table>
### INTERVAL Options

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEMIMONTH</td>
<td>DATE or DATETIME</td>
<td>1st and 16th of each month</td>
<td>DATE9.</td>
</tr>
<tr>
<td>MONTH</td>
<td>DATE or DATETIME</td>
<td>month</td>
<td>MONYY7.</td>
</tr>
<tr>
<td>QUARTER</td>
<td>DATE or DATETIME</td>
<td>3 months</td>
<td>YYQC6.</td>
</tr>
<tr>
<td>SEMIYEAR</td>
<td>DATE or DATETIME</td>
<td>6 months</td>
<td>MONYY7.</td>
</tr>
<tr>
<td>YEAR</td>
<td>DATE or DATETIME</td>
<td>year</td>
<td>YEAR4.</td>
</tr>
</tbody>
</table>

**Default**: AUTO. An appropriate interval is chosen based on the data and the column date, date-time, or time format.

**Restriction**: This option applies to time axes only.

**Requirement**: The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can specify only AUTO, SECOND, MINUTE, HOUR.

**Interaction**: This option is ignored if the TICKVALUELIST= option is used.

**INTERVALMULTIPLIER=positive-integer**

specifies a multiplier to apply to the time interval that is in effect for the axis.

**Note**: This feature applies to SAS 9.4M3 and to later releases.

<table>
<thead>
<tr>
<th>Default</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Restriction</strong></td>
<td>This option applies to time axes only.</td>
</tr>
<tr>
<td><strong>Interaction</strong></td>
<td>This option is ignored if the TICKVALUELIST= option is used.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>Use the INTERVAL= option to specify a different time interval.</td>
</tr>
<tr>
<td><strong>Examples</strong></td>
<td>To specify 3-month intervals:</td>
</tr>
<tr>
<td></td>
<td>INTERVAL=MONTH INTERVALMULTIPLIER=3</td>
</tr>
<tr>
<td></td>
<td>To specify 10-year intervals:</td>
</tr>
<tr>
<td></td>
<td>INTERVAL=YEAR INTERVALMULTIPLIER=10</td>
</tr>
</tbody>
</table>

**MINORGRID=TRUE | FALSE**

specifies whether grid lines are displayed at the minor tick marks.
Defaults

FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips

The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See

“boolean ” on page 1479 for other Boolean values that you can use.

**MINORGRIDATTRS=**

<table>
<thead>
<tr>
<th>style-element</th>
<th>style-element (line-options)</th>
<th>(line-options)</th>
</tr>
</thead>
</table>

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults

The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction

This option is ignored when MINORTICKS=FALSE.
### Note
When *style-element* is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

### Tip
Use the GRIDATTRS= option to control the appearance of the major grid lines.

### See
- “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.
- “Line Options” on page 1522 for available line options.

### Example
Here is an example that specifies light blue, dotted lines for the minor grid.
```plaintext
minorgridattrs=(color=lightblue pattern=dot);
```

---

**MINORTICKINTERVAL=** *interval*

specifies the time interval between minor ticks. See Table 8.8 on page 1040 for information about the intervals that you can select. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, then you must specify AUTO, SECOND, MINUTE, or HOUR.

**Default**
AUTO

**Interactions**
This option is ignored if the TICKVALUELIST= option is used.

This option is ignored if the MINORTICKINTERVAL= setting is greater than the INTERVAL= setting.

---

**MINORTICKS=** TRUE | FALSE

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

![Minor Ticks Example](image)

**Default**
FALSE

**Interactions**
The number of minor ticks is dependent on the value of the MINORTICKINTERVAL= option, if specified. If MINORTICKINTERVAL= is not specified, then it is dependent on the value of the INTERVAL= option.

This option is ignored if the TICKVALUELIST= option is used, or if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick marks.

If MINORTICKS=TRUE is specified with the TICKVALUELIST= option, then MINORTICKS=TRUE is ignored, and minor tick marks are not displayed.

**Tip**
Use the MINORGRID= option to display grid lines at the minor tick values.
See “boolean” on page 1479 for other Boolean values that you can use.

**SPLITTICKVALUE=TRUE | FALSE**

specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available for a Y or Y2 axis.

**TRUE**

splits the axis tick values into two lines allowing more tick values to appear.

For example, with `INTERVAL=MONTH`, this is how tick values are split:

```
Jan Apr Jul Oct Jan Apr Jul Oct Jan
2003 2004 2005 2006
```

**FALSE**

does not split the axis tick values.

For example, when this option specifies FALSE, this is how the tick values in the previous example appear:

```
```

Typically, fewer tick values fit, causing thinning, rotation, or staggering of the values. See the **TICKVALUEFITPOLICY** option.

**Default**  TRUE

**Restriction**  This option applies to time axes only.

**Interaction**  This option is ignored if the `TICKVALUELIST=` or `TICKVALUEFORMAT=` option is used.

See “boolean” on page 1479 for other Boolean values that you can use.

**TICKVALUEFITPOLICY=** *policy*

specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available for the Y and Y2 axes. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. The following policies are valid:

**EXTRACT**

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

**Requirement**  The **EXTRACT** policy must be used with an `AXISLEGEND` statement. For more information, see "Extracting Discrete Axis Tick Values into a Legend" in *SAS Graph Template Language: User’s Guide*.

**EXTRACTALWAYS**

same as **EXTRACT**, except that the extraction is implemented regardless of whether collision occurs.
Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE
makes no attempt to avoid collisions between tick values.
Tick values are displayed even when they collide.

SPLIT
splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.
A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See TICKVALUESPLITCHAR=

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit the available space.

THIN
eliminates alternate tick values.

EXTRACT
display consecutive integers along the axis instead of the actual tick values to represent those tick values.
In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.
NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The **TICKVALUEROTATION=** option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The **TICKVALUEROTATION=** option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the **ROTATEALWAYS** policy, and then drops the tick values if collisions still occur.

ROTATETHIN
attempts the **ROTATE** policy, and then the **THIN** policy.

SPLIT
splits the tick value at a split character, which is specified by the **TICKVALUESPLITCHAR=** option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See **TICKVALUESPLITCHAR=**

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the **TICKVALUESPLITCHAR=** option.

See **TICKVALUESPLITCHAR=**

SPLITROTATE
attempts the **SPLIT** policy, and then the **ROTATE** policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

Note: This feature applies to **SAS 9.4M5** and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

Note: This feature applies to **SAS 9.4M5** and to later releases.

STAGGER
alternates the tick values between two rows.
STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.

TRUNCATETHIN
attempts the TRUNCATE policy, and then the THIN policy.

Default	THIN

Restriction	This option is valid only for the X and X2 axes.

Interaction	When SPLITTICKVALUE= TRUE, this option is ignored and only the THIN policy is used.

Note	A note is written to the SAS log when tick value thinning occurs.

**TICKVALUEFORMAT=format | DATA**
specifies how to format the values for major tick marks.

*format*
specifies a SAS date, time, or datetime format to control how the major tick values are displayed.

This format must be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME and should be appropriate for the value of the INTERVAL= option. For example, if INTERVAL=MONTH and there are two years of data displayed on the axis, then choosing TICKVALUEFORMAT=YEAR. would result in several ticks having the same year value.

*DATA*
specifies that the SAS date, time, or datetime format associated with the data column assigned to the axis be used to control how the major tick values are displayed.

Default	The default format used by the INTERVAL= option. The default does not apply if TICKVALUELIST= is specified.

Restrictions	This option applies to time axes only.

GTL currently honors most but not every SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.
**TICKVALUELIST**=(time-constant-list | date-constant-list | datetime-constant-list | numeric-list)

specifies the tick values for a time axis as list.

**Default**
An internal algorithm determines the tick values.

**Restrictions**
This option applies to time axes only.

If **TICKVALUEPRIORITY**= is set to FALSE, then this option does not extend the data range of the axis. If the values fall within the default data range or that specified by the **VIEWMIN**= or **VIEWMAX**= options, then they are used.

**Requirement**
The tick values must be specified as a space-separated list of values enclosed in parentheses. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

**Interactions**
The values in the list are formatted according to the format specified on the **TICKVALUEFORMAT**= option. If **TICKVALUEFORMAT**= is not used, then the values are formatted according to the column format (the default **TICKVALUEFORMAT** value is not applied to these values).

If this option is specified, the **SPLITTICKVALUE**= and **INTERVAL**= options are ignored.

If MINORTICKS=TRUE is specified with the **TICKVALUELIST**= option, MINORTICKS=TRUE is ignored, and minor tick marks are not displayed.

**TICKVALUEPRIORITY**=TRUE | FALSE

specifies whether the **TICKVALUELIST**= specification can extend the axis data range.

**TRUE**
extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by the **TICKVALUELIST**= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

**FALSE**
displays only the tick values that are specified by the **TICKVALUELIST**= option that fall within the explicit data range set by the **VIEWMIN**= and **VIEWMAX**= options or by the implicit data range set by the actual data minimum and data maximum.

**Default**
**FALSE**

**Interactions**
When this option is set to **TRUE**, the **VIEWMIN**= and **VIEWMAX**= options are ignored.

This option is ignored if the **DISPLAY**= option or the **DISPLAYSECONDARY**= option does not display the tick values.
This option is ignored if the TICKVALUELIST= option is not specified.

This option is ignored when the INCLUDERANGES= option is specified.

Note: If the minimum and maximum of the specified values are within the data range, then this option has no effect.

See “boolean” on page 1479 for other Boolean values that you can use.

**TICKVALUEROTATION=**DIAGONAL | DIAGONAL2 | VERTICAL

specifies how the tick values are rotated on the X and X2 axes.

**DIAGONAL**

rotates the tick values to a 45-degree diagonal position.

The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

**DIAGONAL2**

rotates the tick values to a –45-degree diagonal position.

The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

Note: This feature applies to SAS 9.4M5 and to later releases.

**VERTICAL**

rotates the labels to a 90-degree vertical position.

The labels are always drawn from bottom to top.

**Default**

DIAGONAL

**Restriction**

This option is valid for XAXISOPTS= and X2AXISOPTS= only.

**Interaction**

The TICKVALUEFITPOLICY= option must be set to ROTATE or ROTATEALWAYS for this option to have any effect.

**VIEWMAX=**number

specifies the maximum data value to include in the display.

**Default**

The maximum value in the data for the specified axis.

**Interactions**

This option is ignored when TICKVALUEPRIORITY= TRUE.

This option is ignored when the INCLUDERANGES= option is specified.

**Notes**

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.
Tip
To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

See
“Adjusting the Axis View” on page 983

**VIEWMIN=number**
specifies the minimum data value to include in the display.

**Default**
The minimum value in the data for the specified axis.

**Interactions**
This option is ignored when TICKVALUEPRIORITY= TRUE.

This option is ignored when the INCLUDERANGES= option is specified.

**Notes**
Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip
To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

See
“Adjusting the Axis View” on page 983

**Details**
The LAYOUT OVERLAY statement provides the XAXISOPTS=, YAXISOPTS=, X2AXISOPTS=, Y2AXISOPTS= options that enable you to manage the axis display separately for the X, Y, X2, and Y2 axes. The following example template uses the YAXISOPTS= option to manage the grid lines, tick marks, and tick values on a Y axis:

```plaintext
begingraph;
  layout overlay /
    yaxisopts=(
      griddisplay=on
      display=(ticks tickvalues)
    );
  seriesplot x=month y=predict;
endlayout;
endgraph;
```

Within an OVERLAY layout block, each plot axis is always of a particular type. In the default cases, the axis type is always DISCRETE, LINEAR, or TIME. The **TYPE=** option enables you to specify an axis type that overrides the default. For example, when appropriate for the data, you can request a LOG axis. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types: DISCRETEOPTS=, LINEAROPTS=, LOGOPTS=, and TIMEOPTS=. One or more of these options
can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

Axis Options for LAYOUT OVERLAY3D

Axis options for the plots within an OVERLAY3D layout.

Restriction: The OVERLAY3D layout supports linear and time axes only.

Note: Unless otherwise indicated in an option description, each axis option is available for the X, Y, and Z axis.

See: “LAYOUT OVERLAY3D Statement” on page 166

Syntax

Axis options for the plots within an OVERLAY3D layout are specified with the following options on a LAYOUT OVERLAY3D statement:

- `XAXISOPTS=(axis-options)`
- `YAXISOPTS=(axis-options)`
- `ZAXISOPTS=(axis-options)`

General Options for All Axes in an Overlay3D

The options that are documented in this section can be used with either axis type that is supported within an OVERLAY3D layout. Subsequent sections in the chapter document the axis options that are available only for the specific axis type: linear or time.

<table>
<thead>
<tr>
<th>Statement Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Controls which axis features are displayed.</td>
</tr>
<tr>
<td>GRIDATTRS</td>
<td>Specifies the attributes of the grid lines.</td>
</tr>
<tr>
<td>GRIDDISPLAY</td>
<td>Specifies whether axis grid lines are displayed.</td>
</tr>
<tr>
<td>LABEL</td>
<td>Specifies the axis label.</td>
</tr>
<tr>
<td>LABELATTRS</td>
<td>Specifies the color and font attributes of the axis label.</td>
</tr>
<tr>
<td>LINEAROPTS</td>
<td>Specifies options for a standard numeric interval axis.</td>
</tr>
<tr>
<td>OFFSETMAX</td>
<td>Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>OFFSETMIN</td>
<td>Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>Statement Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TICKVALUEATTRS</td>
<td>Specifies the color and font attributes of the axis tick values.</td>
</tr>
<tr>
<td>TIMEOPTS</td>
<td>Specifies options for a TIME axis.</td>
</tr>
<tr>
<td>TYPE</td>
<td>Specifies the type of axis to use.</td>
</tr>
</tbody>
</table>

**DISPLAY=STANDARD | ALL | NONE | (display-options)**

controls which axis features are displayed on the primary axis.

- **STANDARD**
  - specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed
- **ALL**
  - currently the same as STANDARD
- **NONE**
  - specifies that no axis features are displayed

*(display-options)*

- specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

- **LABEL**
  - displays the axis label
- **LINE**
  - displays the axis line
- **TICKS**
  - displays the tick marks
- **TICKVALUES**
  - displays the values that are represented by the major tick marks

**Default**

STANDARD

**Tips**

The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

When LINE is excluded from the DISPLAY= option, the layout wall outline or the default baseline of a bar chart, needle plot, or waterfall chart can appear to be an axis line. To suppress the wall outline, use the WALLDISPLAY= option in the layout statement. To suppress the plot baseline, use the BASELINEATTRS= option in the plot statement.

**GRIDATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the grid lines.

**Default**

The GraphGridLines style element.

**Interaction**

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.
See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**GRIDDISPLAY=** **AUTO_OFF** | **AUTO_ON** | **ON** | **OFF**

specifies whether axis grid lines are displayed. This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

**AUTO_OFF**

specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

**AUTO_ON**

specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

**ON**

specifies that grid lines are always displayed.

The current style has no override.

**OFF**

specifies that grid lines are never displayed.

The current style has no override.

The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

<table>
<thead>
<tr>
<th>GRIDDISPLAY= option</th>
<th>DisplayOpts= style attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_OFF</td>
<td>AUTO</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>AUTO</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>ON</td>
<td>any value</td>
<td>yes</td>
</tr>
<tr>
<td>OFF</td>
<td>any value</td>
<td>no</td>
</tr>
</tbody>
</table>

Default: AUTO_OFF
Note Supplied styles use DisplayOpts="AUTO," which means that the style has no preference about grid lines and the graphics template setting for grid lines is always used.

**LABEL=**"string" | ("string" ..."string")
specifies the axis label. The string can be either a string literal or a dynamic. The list form implies that all included string literals or dynamic variables will be concatenated.

**Default** The default label is derived from the primary plot in the layout. For more information, see “When Plots Share Data and a Common Axis” on page 979.

**Interaction** This option is ignored if the DISPLAY= option does not display the axis label.

**Note** If the axis label is too long to fit along the axis, then it is truncated by default.

**Tip** Use the ODS escape sequence to specify Unicode characters in an axis label. Here is an example that specifies degrees Fahrenheit in an axis label for temperature:

```plaintext
label="Temperature ((*ESC*){unicode '00B0'x}F)"
```

See “Overriding the Default Axis Label” in SAS Graph Template Language: User’s Guide

**LABELATTRS=**style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the axis label.

**Default** The GraphLabelText style element.

**Interaction** This option is ignored if the DISPLAY= option does not display the axis label.

**See** “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element. “Text Options” on page 1525 for available text-options.

**LINEAROPTS=(linear-axis-options)**
specifies one or more options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

**Interaction** This option is ignored if the axis type is not LINEAR.

**See** “Options for Linear Axes Only” on page 1056 for the options that you can use for linear-axis-options.

**OFFSETMAX=**AUTO | number
reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.

**AUTO** reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.
number
specifies the offset as a decimal proportion of the full axis length.
For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater.

Default AUTO
Range 0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.
See “Adjusting Axis Offsets” on page 985

OFFSETMIN= AUTO | number
reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.

AUTO
reserves just enough area to fully display markers and other graphical features near the minimum end of an axis.

number
specifies the offset as a decimal proportion of the full axis length.
For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less.

Default AUTO
Range 0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.
See “Adjusting Axis Offsets” on page 985

TICKVALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the axis tick values.

Default The GraphValueText style element.

Interaction This option is ignored if the DISPLAY= option does not display tick values.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

TIMEOPTS=(time-axis-options)
specifies one or more options for a time axis.

Requirements Columns associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction This option is ignored if the axis type is not TIME.
See “Options for Time Axes Only” on page 1064 for the options that you can use for time-axis-options.

**TYPE=** AUTO | LINEAR | TIME

specifies the type of axis to use.

**AUTO**

requests that the axis type be automatically determined by the plot or the overlay contents.

**LINEAR**

uses a LINEAR axis if possible.

You can add a LINEAROPTS= option list to customize this axis type.

**TIME**

uses a TIME axis if possible.

Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPS= option list to customize this axis type.

**Default** AUTO

**Interactions**

If this option is set to anything other than AUTO, then plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

After the axis type is determined (whether you set a specific type or AUTO is in effect), only options supported by that axis type can be used. For example, if TYPE=TIME, then only the general OVERLAY3D axis options and those available on TIMEOPS= are supported.

### Options for Linear Axes Only

This section documents the options that can be used with the LINEAROPTS= axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Linear Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>Specifies that evenly spaced integer values are used for tick marks.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKCOUNT</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
</tbody>
</table>
### Linear Axis Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>THRESHOLDMAX</td>
<td>Specifies a bias for including one more tick mark at the maximum end of the axis.</td>
</tr>
<tr>
<td>THRESHOLDMIN</td>
<td>Specifies a bias for including one more tick mark at the minimum end of the axis.</td>
</tr>
<tr>
<td>TICKDISPLAYLIST</td>
<td>Specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values for a linear axis as list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.</td>
</tr>
<tr>
<td>TICKVALUESEQUENCE</td>
<td>Specifies the tick values for a linear axis by start, end, and increment.</td>
</tr>
</tbody>
</table>

**INTEGER=TRUE | FALSE**

specifies that evenly spaced integer values are used for tick marks.

- **Default**: FALSE

- **Interactions**: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

  This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option.

  INTEGER=TRUE is ignored for the X or X2 axis when a histogram plot is the primary plot and BINAXIS=TRUE is specified in the HISTOGRAM or HISTOGRAMPARM statement.

- **See**: “boolean” on page 1479 for other Boolean values that you can use.

**MINORGRID=TRUE | FALSE**

specifies whether grid lines are displayed at the minor tick marks.
Defaults
FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction
This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips
The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See
“boolean ” on page 1479 for other Boolean values that you can use.

MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults
The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction
This option is ignored when MINORTICKS=FALSE.
Note When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip Use the GRIDATTRS= option to control the appearance of the major grid lines.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example Here is an example that specifies light blue, dotted lines for the minor grid.

```plaintext
minorgridattrs=(color=lightblue pattern=dot);
```

**MINORTICKCOUNT=** positive-integer

specifies the number of minor ticks that are displayed on the axis.

Defaults Four ticks with five intervals in SAS 9.4M1 and earlier releases.

One tick with two intervals starting with SAS 9.4M2.

Interactions The DISPLAY= option specification must include Ticks for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

Tip To display $n$ intervals between major ticks, use MINORTICKCOUNT=$n-1$.

**MINORTICKS=** TRUE | FALSE

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

![Minor Ticks Diagram]

Default FALSE

Interactions This option is ignored if the TICKVALUELIST= option is used or if the tick marks are not enabled by the DISPLAY= option.

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Tip Use the MINORGRID= option to display grid lines at the minor tick values.

See “boolean” on page 1479 for other Boolean values that you can use.
THRESHOLDMAX=number
specifies a bias for including one more tick mark at the maximum end of the axis.

Default 0.30
Range 0–1
Restriction This option applies to linear axes only.
Interaction This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.
Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.
Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.
Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.
For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0.
See “Adjusting Axis Thresholds” on page 984

THRESHOLDMIN=number
specifies a bias for including one more tick mark at the minimum end of the axis.

Default 0.30
Range 0–1
Restriction This option applies to linear axes only.
Interaction This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.
Tips If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.
Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.
Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.
For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0.
See “Adjusting Axis Thresholds” on page 984

TICKDISPLAYLIST=(string-list)
specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the TICKVALUELIST= option. The strings map one-to-one positionally with the values that are listed in the TICKVALUELIST= option.
The display of tick values is controlled by the `TICKVALUEFORMAT=` option.

**Requirements**

The list of values must be enclosed in parentheses.

Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.

**Interaction**

When this option is specified, the `TICKVALUEFORMAT=` option is ignored.

**Tip**

This option should be used with the `TICKVALUELIST=` option. The number of items in the list for this option should equal the number of items in the list for the `TICKVALUELIST=` option.

**TICKVALUEFORMAT=(format-options) | DATA | format**

specifies how to format the values for major tick marks.

*format-options*

specifies one or more formatting options for major tick values.

Together, these options provide parameters for determining an optimal format (w.d, Ew., BESTw.) for displaying major tick values.

**MAXWIDTH=integer**

specifies the maximum width for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default: 8

**MAXDECIMALS=integer**

specifies the maximum number of decimals for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default: 6

**Note**

The MAXWIDTH= option value should be greater than the MAXDECIMALS= option value.

**PREFERREDDECIMALS=integer**

specifies the number of decimal places that you want to display for most values.

The actual number might vary based on other constraints.

Default: 2

**EXTRACTSCALE=TRUE | FALSE**

specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.

The scale can be a named scale or a scientific-notation scale. The `EXTRACTSCALETYPE=` option specifies the scale type. The scale that is used is appended to the axis label, as shown in the following example.

Total Sales (millions)

For long axis labels, if the scale does not fit the available space, then the label is truncated, and the scale is appended to the truncated label.
Ellipses indicate that the label was truncated, as shown in the following example.

**Total Sales for the Fourth Quarter Of ... (millions)**

In extreme cases in which the scale does not fit even with truncation, the entire axis is dropped.

**Default**  
FALSE

**Restriction**  
The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale for all locales.

**Interactions**  
The scale type is determined by the EXTRACTSCALETYPE= option.

If the axis label is not displayed, then the EXTRACTSCALE=TRUE option is ignored.

**Note**  
When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

**See**  
“boolean” on page 1479 for other Boolean values that you can use.

**EXTRACTSCALETYPE**=DEFAULT | SCIENTIFIC  
specifies whether to extract a named scale or a scientific-notation scale.

**DEFAULT**  
extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as \(10^n\)) for values over 999 trillion. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (\(10^{-n}\)) for values less than 1 trillionth.

**SCIENTIFIC**  
extracts a scientific-notation scale.

A scientific-notation scale is a multiple of 10 expressed as \(10^n\) for values greater than 1, or a multiple of 1/10 expressed as \(10^{-n}\) for values less than 1.

**Default**  
DEFAULT

**Restriction**  
The scale is derived from the English locale for all locales.

**DATA**  
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

**format**  
specifies a format to apply to the major tick values.
Restriction  GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

Note  If you specify a format that significantly reduces precision, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

Default  (MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE, EXTRACTSCALETYPE=DEFAULT)

Interaction  This option is ignored when the TICKDISPLAYLIST= option is specified.

**TICKVALUELIST=(numeric-list)**  
specifies the tick values for a linear axis as a list.

Default  An internal algorithm determines the tick values.

Restriction  This option applies to linear axes only.

Requirement  The tick values must be specified as a space-separated list of numeric values, enclosed in parentheses.

Interactions  This option overrides the INTEGER= option.

This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the DISPLAY= option does not display tick values.

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Tip  The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

**TICKVALUEPRIORITY=TRUE | FALSE**  
specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

**TRUE**  
extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

**FALSE**  
displays only the tick values that are specified by the TICKVALUELIST= option that fall within the explicit data range set by the actual data minimum and data maximum.

Default  FALSE
Restriction
This option applies to linear axes only.

Interactions
This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is not specified.
This option is ignored if the DISPLAY= option does not display the tick values.

Note
If the minimum and maximum of the specified values are within the data range, then this option has no effect.

See
"boolean " on page 1479 for other Boolean values that you can use.

**TICKVALUESEQUENCE=(START=number END=number INCREMENT=number)** specifies the tick values by start, end, and increment.

**START=number**
specifies the value for the first tick mark.

**END=number**
specifies the value for the last tick mark.

**INCREMENT=number**
specifies the increment for intermediate tick marks between the first and last tick marks.

The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

Default
An internal algorithm determines the tick marks.

Interactions
This option overrides the INTEGER= option.

If TICKVALUEPRIORITY= TRUE, then the tick sequence might extend the explicit data range of the axis, but never reduce it.

This option is ignored if the DISPLAY= option does not display tick marks.

Tip
The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

See
TICKVALUELIST= option as an alternative for customizing tick marks.

Options for Time Axes Only
This section documents the options that can be used with the TIMEOPTS= axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Time Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
<td>Specifies the time interval between major tick marks.</td>
</tr>
<tr>
<td>INTERVALMULTIPLIER</td>
<td>Specifies a multiplier to apply to the time interval that is in effect for the axis.</td>
</tr>
</tbody>
</table>
### Time Axis Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKINTERVAL</td>
<td>Specifies the time interval between minor ticks.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values for a time axis as list.</td>
</tr>
</tbody>
</table>

#### INTERVAL=AUTO | SECOND | MINUTE | HOUR | DAY | TENDAY | WEEK | SEMIMONTH | MONTH | QUARTER | SEMIYEAR | YEAR

specifies the time interval between major ticks.

**Table 8.2  Time Intervals**

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>DATE, TIME, or DATETIME</td>
<td>automatically chosen</td>
<td>automatically chosen</td>
</tr>
<tr>
<td>SECOND</td>
<td>TIME or DATETIME</td>
<td>second</td>
<td>TIME8.</td>
</tr>
<tr>
<td>MINUTE</td>
<td>TIME or DATETIME</td>
<td>minute</td>
<td>TIME8.</td>
</tr>
<tr>
<td>HOUR</td>
<td>TIME or DATETIME</td>
<td>hour</td>
<td>TIME8.</td>
</tr>
<tr>
<td>DAY</td>
<td>DATE or DATETIME</td>
<td>day</td>
<td>DATE9.</td>
</tr>
<tr>
<td>TENDAY</td>
<td>DATE or DATETIME</td>
<td>10 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>WEEK</td>
<td>DATE or DATETIME</td>
<td>7 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>SEMIMONTH</td>
<td>DATE or DATETIME</td>
<td>1st and 16th of each month</td>
<td>DATE9.</td>
</tr>
<tr>
<td>MONTH</td>
<td>DATE or DATETIME</td>
<td>month</td>
<td>MONYY7.</td>
</tr>
</tbody>
</table>
## INTERVAL Option

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value format</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUARTER</td>
<td>DATE or DATETIME</td>
<td>3 months</td>
<td>YYQC6.</td>
</tr>
<tr>
<td>SEMIYEAR</td>
<td>DATE or DATETIME</td>
<td>6 months</td>
<td>MONYY7.</td>
</tr>
<tr>
<td>YEAR</td>
<td>DATE or DATETIME</td>
<td>year</td>
<td>YEAR4.</td>
</tr>
</tbody>
</table>

**Default**: AUTO. An appropriate interval is chosen based on the data and the column date, date-time, or time format.

**Restriction**: This option applies to time axes only.

**Requirement**: The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can specify only AUTO, SECOND, MINUTE, HOUR.

**Interaction**: This option is ignored if the TICKVALUELIST= option is used.

### INTERVALMULTIPLIER=positive-integer

specifies a multiplier to apply to the time interval that is in effect for the axis.

**Default**: 1

**Restriction**: This option applies to time axes only.

**Interaction**: This option is ignored if the TICKVALUELIST= option is used.

**Tip**: Use the INTERVAL= option to specify a different time interval.

**Examples**

To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULTIPLIER=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULTIPLIER=10
```

### MINORGRID=TRUE | FALSE

specifies whether grid lines are displayed at the minor tick marks.
Defaults
FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpt style reference starting with SAS 9.4M2. If attribute DisplayOpt is not defined in the active style, then FALSE is the default value.

Interaction
This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips
The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See “boolean” on page 1479 for other Boolean values that you can use.

MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults
The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction
This option is ignored when MINORTICKS=FALSE.
Note  When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip  Use the GRIDATTRS= option to control the appearance of the major grid lines.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example  Here is an example that specifies light blue, dotted lines for the minor grid.

    minorgridattrs=(color=lightblue pattern=dot);

MINORTICKINTERVAL=\textit{interval}

specifies the time interval between minor ticks. See Table 8.9 on page 1065 for information about the intervals that you can select. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, then you must specify AUTO, SECOND, MINUTE, or HOUR.

Default  AUTO

Interactions  This option is ignored if the TICKVALUELIST= option is used.

This option is ignored if the MINORTICKINTERVAL= setting is greater than the INTERVAL= setting.

MINORTICKS=\texttt{TRUE} | \texttt{FALSE}

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

\begin{figure}
\centering
\begin{tabular}{cccccc}
\hline
\textbf{Jan} & \textbf{Feb} & \textbf{Mar} & \textbf{Apr} & \textbf{May} & \textbf{Jun} \\
\hline
\end{tabular}
\end{figure}

Default  FALSE

Interactions  The number of minor ticks is dependent on the value of the MINORTICKINTERVAL= option, if specified. If MINORTICKINTERVAL= is not specified, then it is dependent on the value of the INTERVAL= option.

This option is ignored if the TICKVALUELIST= option is used or if the tick marks are not enabled by the DISPLAY= option.

If MINORTICKS=TRUE is specified with the TICKVALUELIST= option, then MINORTICKS=TRUE is ignored, and minor tick marks are not displayed.

Tip  Use the MINORGRID= option to display grid lines at the minor tick values.

See  “\texttt{boolean }” on page 1479 for other Boolean values that you can use.
**TICKVALUEFORMAT=** *format* | **DATA**
specifies how to format the values for major tick marks.

*format*
specifies a SAS date, time, or datetime format to control how the major tick values are displayed.

This format must be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME and should be appropriate for the value of the INTERVAL= option. For example, if INTERVAL=MONTH and there are two years of data displayed on the axis, then choosing TICKVALUEFORMAT=YEAR would result in several ticks having the same year value.

**DATA**
specifies that the SAS date, time, or datetime format associated with the data column assigned to the axis be used to control how the major tick values are displayed.

**Default**
The default format used by the INTERVAL= option. The default does not apply if TICKVALUELIST= is specified.

**Restrictions**
This option applies to time axes only.

GTL currently honors most but not every SAS format. For details, see Appendix 5, "SAS Formats Not Supported," on page 1541.

**TICKVALUELIST=**(time-constant-list | date-constant-list | datetime-constant-list | numeric-list)
specifies the tick values for a time axis as list.

**Default**
An internal algorithm determines the tick values.

**Restrictions**
This option applies to time axes only.

If TICKVALUEPRIORITY= is set to FALSE, then this option does not extend the data range of the axis. If the values fall within the default data range, then they are used.

**Requirement**
The tick values must be specified as a space-separated list of values enclosed in parentheses. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

**Interactions**
The values in the list are formatted according to the format specified on the TICKVALUEFORMAT= option. If TICKVALUEFORMAT= is not used, then the values are formatted according to the column format (the default TICKVALUEFORMAT value is not applied to these values).

If this option is specified, then the INTERVAL= option is ignored.

If MINORTICKS=TRUE is specified with the TICKVALUELIST= option, MINORTICKS=TRUE is ignored, and minor tick marks are not displayed.
Details

The LAYOUT OVERLAY3D statement provides XAXISOPTS=, YAXISOPTS=, and ZAXISOPTS= options that enable you to manage the axis display separately for the X, Y, and Z axes. The following example template uses the YAXISOPTS= option to manage the display of grid lines, tick marks, and tick values on a Y axis:

```
begingraph;
layout overlay3d /
  yaxisopts=(
    griddisplay=on
    display=(ticks tickvalues)
  );
bihistogram3dparm x=rater y=customer z=percent;
endlayout;
endgraph;
```

Within an OVERLAY3D layout block, each plot axis is always either a linear or a time axis. The default axis-type setting is AUTO, which specifies that the axis type be automatically determined by the plot or the overlay contents. The TYPE= option enables you to explicitly specify either a LINEAR or a TIME axis type. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the axis options LINEAROPTS= and TIMEOPS= enable you to specify features for a linear or a time axis. You can combine one or more general axis options with the options for the specific axis type. However, specified settings are applied only to the axis type that supports them. For example, if you specify general axis options with time axis options and the generated graph produces a linear axis type, then the time axis options are ignored.

Row and Column Axis Options for LAYOUT LATTICE

Row and column axis options for the plots within a LATTICE layout.

See: "LAYOUT LATTICE Statement" on page 118

Syntax

Axis options for the plots in a LATTICE layout are specified with the following statements in a LAYOUT LATTICE statement block:

- `COLUMNAXIS /column-axis-options`
- `ROWAXIS /row-axis-options`

General Options for All Row and Column Axes in a Lattice

The options that are documented in this section can be used with any of the row or column axis types that are supported in a LAYOUT LATTICE statement. Subsequent sections in the chapter document the axis options that are available only for specific axis types: discrete, linear, log, or time axes.
<table>
<thead>
<tr>
<th>Statement Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISCRETEOPTS</td>
<td>Specifies features for a discrete axis.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Controls which axis features are displayed on the primary axis.</td>
</tr>
<tr>
<td>DISPLAYSECONDARY</td>
<td>Controls which axis features are displayed on the secondary axis.</td>
</tr>
<tr>
<td>GRIDATTRS</td>
<td>Specifies the attributes of the grid lines.</td>
</tr>
<tr>
<td>GRIDDISPLAY</td>
<td>Specifies whether axis grid lines are displayed.</td>
</tr>
<tr>
<td>LABEL</td>
<td>Specifies the axis label.</td>
</tr>
<tr>
<td>LABELATTRS</td>
<td>Specifies the color and font attributes of the axis label.</td>
</tr>
<tr>
<td>LABELFITPOLICY</td>
<td>Specifies a policy for fitting axis labels in the available space.</td>
</tr>
<tr>
<td>LABELPOSITION</td>
<td>Specifies the position of the axis label.</td>
</tr>
<tr>
<td>LABELSPLITCHAR</td>
<td>Specifies one or more characters on which the axis labels can be split, if needed.</td>
</tr>
<tr>
<td>LABELSPLITCHARDROP</td>
<td>Specifies whether the split characters should be included in the axis labels that are displayed.</td>
</tr>
<tr>
<td>LABELSPLITJUSTIFY</td>
<td>Specifies the justification of the strings that are inside the axis label blocks.</td>
</tr>
<tr>
<td>LINEAROPTS</td>
<td>Specifies features for a standard numeric interval axis.</td>
</tr>
<tr>
<td>LOGOPTS</td>
<td>Specifies features for a log axis.</td>
</tr>
<tr>
<td>NAME</td>
<td>Assigns a name to an axis for reference in other statements.</td>
</tr>
<tr>
<td>OFFSETMAX</td>
<td>Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>OFFSETMIN</td>
<td>Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>REVERSE</td>
<td>Specifies whether the axis origin should be reversed.</td>
</tr>
<tr>
<td>Statement Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SHORTLABEL</td>
<td>Specifies an alternate axis label to use if the default or specified axis label is too long for the axis length.</td>
</tr>
<tr>
<td>TICKVALUEATTRS</td>
<td>Specifies the color and font attributes of the axis tick values.</td>
</tr>
<tr>
<td>TICKVALUEHALIGN</td>
<td>Specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.</td>
</tr>
<tr>
<td>TICKVALUEVALIGN</td>
<td>Specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.</td>
</tr>
<tr>
<td>TIMEOPTS</td>
<td>Specifies features for a TIME axis.</td>
</tr>
<tr>
<td>TYPE</td>
<td>Specifies the type of axis to use.</td>
</tr>
</tbody>
</table>

**DISCRETEOPTS=(discrete-axis-options)**

Specifies one or more options for a discrete axis. Options must be enclosed in parentheses. Each option is specified as a `name = value` pair and each pair is space separated.

**Interaction**

This option is ignored if the axis type is not DISCRETE.

**See**

“Options for Row and Column Discrete Axes Only” on page 1082 for the options that you can use for `discrete-axis-options`.

**DISPLAY=STANDARD | ALL | NONE | (display-options)**

Controls which axis features are displayed on the primary axis.

- **STANDARD**
  - Specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

- **ALL**
  - Currently the same as STANDARD

- **NONE**
  - Specifies that no axis features are displayed

**display-options**

Specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

- **LABEL**
  - Displays the axis label

- **LINE**
  - Displays the axis line

- **TICKS**
  - Displays the tick marks

- **TICKVALUES**
  - Displays the values that are represented by the major tick marks
The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

When LINE is excluded from the DISPLAY= option, the layout wall outline or the default baseline of a bar chart, needle plot, or waterfall chart can appear to be an axis line. To suppress the wall outline, use the WALLDISPLAY= option in the layout statement. To suppress the plot baseline, use the BASELINEATTRS= option in the plot statement.

"Details" on page 1124 for more information about the primary and secondary axes.

**DISPLAYSECONDARY** = NONE | ALL | STANDARD | (display-options)

controls which axis features are displayed on the secondary axis. A secondary axis is not an independent axis. Rather, it mirrors the primary axis. Thus, for this option to take effect, all plot statements in the layout must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left).

- **NONE** specifies that no axis features are displayed.
- **STANDARD** specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis.
- **ALL** currently the same as STANDARD

(display-options)

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

- **LABEL** displays the axis label
- **LINE** displays the axis line
- **TICKS** displays the tick marks
- **TICKVALUES** displays the values that are represented by the major tick marks

If some plot statements set XAXIS=X and others set XAXIS=X2, both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y axes.
Interactions

This option is ignored if the COLUMNAXIS statement appears within a COLUMNAXES block and COLUMN2DATARANGE=UNION or UNIONALL is in effect.

This option is ignored if the COLUMNAXIS statement appears within a COLUMN2AXES block and COLUMNDATARANGE=UNION or UNIONALL is in effect.

Tip

Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

See

"Details" on page 1124 for more information about the primary and secondary axes.

GRIDATTRS=style-element | style-element (line-options) | (line-options)

specifies the attributes of the grid lines.

Default

The GraphGridLines style element.

Interaction

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tip

On a log axis, this option affects the appearance of the major grid lines only. It does not affect the appearance of the minor grid lines. To control the appearance of the minor grid lines on a log axis, use the MINORGRIDATTRS= option.

See

"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Line Options" on page 1522 for available line-options.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF

specifies whether axis grid lines are displayed. This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed. When displayed, the grids appear in all cells.

AUTO_OFF

specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON

specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON

specifies that grid lines are always displayed.

The current style has no override.

OFF

specifies that grid lines are never displayed.

The current style has no override.

The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.
<table>
<thead>
<tr>
<th>GRIDDISPLAY= option</th>
<th>DisplayOpts= style attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_OFF</td>
<td>AUTO</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>AUTO</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>ON</td>
<td>any value</td>
<td>yes</td>
</tr>
<tr>
<td>OFF</td>
<td>any value</td>
<td>no</td>
</tr>
</tbody>
</table>

**Default**  AUTO_OFF

**Note** Supplied styles use DisplayOpts="AUTO," which means that the style has no preference about grid lines and the graphics template setting for grid lines is always used.

**LABEL=** "string" | ("string" ..."string")
specifies the axis label. The string can be either a string literal or a dynamic. The list form implies that all included string literals or dynamic variables will be concatenated.

**Default** The default label is derived from the primary plot in the layout. For more information, see “When Plots Share Data and a Common Axis” on page 979.

**Interaction** This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

**Note** If the axis label is too long to fit along the axis, then it is truncated by default.

**Tips** Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Use the ODS escape sequence to specify Unicode characters in an axis label. Here is an example that species degrees Fahrenheit in an axis label for temperature:

```
label="Temperature ((*ESC*){unicode '00B0'x}F)"
```

**See** “Overriding the Default Axis Label” in SAS Graph Template Language: User’s Guide

**LABELATTRS=** style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the axis label.
The GraphLabelText style element.

This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**LABELFITPOLICY=**AUTO | SPLITALWAYS | STACKED

specifies a policy for fitting axis labels in the available space.

**AUTO**

uses the short label, when specified, instead of the original label.

If the short label does not fit, then it is clipped. When no short label is specified, the original label is clipped.

**SPLIT**

splits the axis label at a split character only when necessary in order to make the label fit the available space.

The split character is specified by the LABELSPLITCHAR= option. The short label is not used. A split does not occur at a split character if a split is not needed at that location. If the label does not contain any of the specified split characters, then it is not split. A label that cannot be split or that does not fit the available space even after splitting might overlap the adjoining space.

**SPLITALWAYS**

always split the axis label at every occurrence of a split character, which is specified by the LABELSPLITCHAR= option.

If the label cannot be split, then it is clipped.

**STACKED**

displays the Y or Y2 axis label vertically as stacked letters.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**Valid in** ROWAXIS= only

**Default** AUTO

This option has effect only when LABELPOSITION= is CENTER or DATACENTER.

**Note**

When LABELPOSITION=CENTER, the available area is the full axis, including the axis offsets. When LABELPOSITION=DATACENTER, the available area is the tick display area, excluding the axis offsets.

**LABELPOSITION=**CENTER | DATACENTER

specifies the position of the axis label.

**CENTER**

centers each row or column axis label in its axis area.

For the row Y and Y2 axes, the label is oriented vertically and is centered in the axis area (including the offsets) of its row. It is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis. For
the column X and X2 axes, the label is centered in the axis area (including the offsets) of its column. It is positioned below the tick values for the X axis or above the axis values for the X2 axis.

**DATACENTER**

centers each row or column axis label in its axis tick display area.

For the row Y and Y2 axes, each label is oriented vertically and is centered in the axis tick display area (excluding the offsets) of its row. The labels are positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis. For the column X and X2 axes, each label is centered in the axis tick display area (excluding the offsets) of its column. The labels are positioned below the tick values for the X axis or above the axis values for the X2 axis.

The following figure shows the CENTER and DATACENTER positions for row Y-axis labels Open and Close, and column X-axis label Month. An offset is applied to the maximum end of each axis for demonstration purposes.

**LABELSPLITCHAR=**"character-list"

specifies one or more characters on which the axis labels can be split, if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the axis label. In that case, all of the specified split characters together are treated as a single split character.

When LABELFITPOLICY=SPLIT, if the axis label does not fit the available space, then it is split on a specified split character only if a split is needed at that point to make the label fit. In this case, a split might not occur on every split character. When LABELFITPOLICY=SPLITALWAYS, the axis label is split unconditionally on every occurrence of a split character. If the axis label does not contain any of the specified split characters, the label is not split.

"character-list"

one or more characters with no space between each character and enclosed in quotation marks.

Default A blank space

Requirements The list of characters must be enclosed in quotation marks.
Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

```
labelsplitchar="abc"
```

**Interactions**

This option has effect only when `LABELPOSITION=` is `CENTER` or `DATACENTER`.

The `LABELSPLITCHARDROP=` option specifies whether the split characters are included in the displayed data label or are dropped.

**Notes**

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

**Tip**

Use the `LABELSPLITJUSTIFY=` option to specify the justification of the strings in the axis label block.

**LABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the displayed axis labels.

- **TRUE**
  - drops the split characters from the axis label display.

- **FALSE**
  - includes the split characters in the axis label display.

  When the label is split while `LABELSPLITCHARDROP=FALSE` is in effect, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

**Default**

`TRUE`. The split characters are dropped from the axis label.

**Interactions**

This option has effect only when `LABELPOSITION=` is `CENTER` or `DATACENTER`.

The `LABELSPLITCHAR=` option specifies the split characters.

**See**

“`boolean`” on page 1479 for other Boolean values that you can use.

**LABELSPLITJUSTIFY=justification**

specifies the justification of the strings that are inside the axis label blocks.

**justification**

- **CENTER**
- **LEFT**
- **RIGHT**

  specifies the justification for the X or X2 axis label.

- **CENTER**
- **TOP**
- **BOTTOM**

  specifies the justification for the Y or Y2 axis label.

**Default**

`CENTER`
Interaction  This option has effect only when LABELPOSITION= is CENTER or DATACENTER.

**LINEAROPTS=(linear-axis-options)**
specifies one or more options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction  This option is ignored if the axis type is not LINEAR.

See  “Options for Row and Column Linear Axes Only” on page 1091 for the options that you can use for linear-axis-options.

**LOGOPTS=(log-axis-options)**
specifies one or more options for a log axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction  This option is ignored if the axis type is not LOG.

See  “Options for Row and Column Log Axes Only” on page 1104 for the options that you can use for log-axis-options.

**NAME="string"**
assigns a name to an axis for reference in other statements. Currently, it is used only in an AXISLEGEND statement.

Restriction  This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interactions  This option is ignored unless the axis is discrete. The axis can be discrete by default, or explicitly set to discrete with a TYPE= DISCRETE setting.

For this option to take effect, an axis legend must be enabled. To enable an axis legend, the DISCRETEOPTS= option must set the TICKVALUEFITPOLICY to either EXTRACT or EXTRACTALWAYS. In addition, an AXISLEGEND statement must be specified to generate the axis legend.

**OFFSETMAX=AUTO | AUTOCOMPRESS | number**
reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.

**AUTO**
reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

**AUTOCOMPRESS**
applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

**number**
specifies the offset as a decimal proportion of the full axis length.

Default  AUTO

Range  0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.
OFFSETMIN=AUTO | AUTOCOMPRESS | *number*
reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.

AUTO
reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

*number*
specifies the offset as a decimal proportion of the full axis length.

Default AUTO

Range 0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

See “Adjusting Axis Offsets” on page 985.

REVERSE=TRUE | FALSE
specifies whether tick values should appear in the reverse order.

Default FALSE

See “boolean ” on page 1479 for other Boolean values that you can use.

SHORTLABEL="*string*"
specifies an alternate axis label to display when the default label or the label specified by the LABEL= option is too long to fit the available space.

Interaction This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

Note If the specified label is itself too long for the axis, it is truncated in the display.

TICKVALUEATTRS=*style-element* | *style-element (text-options)* | (text-options)
specifies the color and font attributes of the axis tick values.

Default The GraphValueText style element.

Interaction This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick values.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element.

“Text Options” on page 1525 for available *text-options*.

TICKVALUEHALIGN=LEFT | CENTER | RIGHT
specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.
Defaults

RIGHT for a Y axis

LEFT for a Y2 axis

Restriction

This option is valid for the Y and Y2 axes only.

**TICKVALUEVALIGN=TOP | CENTER | BOTTOM**

specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.

Defaults

TOP for an X axis

BOTTOM for an X2 axis

Restriction

This option is valid for the X and X2 axes only.

**TIMEOPTS=(time-axis-options)**

specifies one or more options for a time axis.

Requirements

Columns associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction

This option is ignored if the axis type is not TIME.

See

“Options for Row and Column Time Axes Only” on page 1113 for the options that you can use for time-axis-options.
TYPE=AUTO | DISCRETE | LINEAR | TIME | LOG
specifies the type of axis to use.

AUTO
requests that the axis type be automatically determined, based on the overlay contents.

DISCRETE
uses a DISCRETE axis if possible.

The data for discrete axes can be character or numeric. You can add a DISCRETEOPTS= option list to customize this axis type.

LINEAR
uses a LINEAR axis if possible.

You can add a LINEAROPTS= option list to customize this axis type.

TIME
uses a TIME axis if possible.

Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPTS= option list to customize this axis type.

LOG
uses a LOG axis if possible.

You can add a LOGOPTS= option list to customize this axis type.

Interaction
If a log axis is requested and the axis data contains 0 or negative values, the axis reverts to a linear axis. This outcome can occur for the response axis of a bar chart, line chart, needle plot, or waterfall chart when a baseline intercept of 0 or less is specified. It can also occur for the response axis of a waterfall chart when an initial bar value of 0 or less is specified. To get a log response axis in those cases, set the baseline intercept or initial bar value to a positive value.

Default
AUTO

Interactions
If this option is set to anything other than AUTO, then plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see "Plot Axis Types Must Agree on Common Axes" on page 981.

After the axis type is determined (whether you set a specific type or AUTO is in effect), you can use only options that are supported by that axis type. For example, if TYPE=TIME, then only the general OVERLAY axis options and those available on TIMEOPTS= are supported.

Options for Row and Column Discrete Axes Only
The options that are documented in this section can be used with the DISCRETEOPTS= axis option. The following table provides a summary of the options.
<table>
<thead>
<tr>
<th>Discrete Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICKDISPLAYLIST</td>
<td>Specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>TICKTYPE</td>
<td>Specifies the position of the axis tick mark.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision on an axis.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the text that is to be displayed for the tick values that are defined in the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>TICKVALUEROTATION</td>
<td>Specifies how the tick values are rotated on the X and X2 axes.</td>
</tr>
<tr>
<td>TICKVALUESPLITCHAR</td>
<td>Specifies a list of characters on which the tick values can be split, if needed.</td>
</tr>
<tr>
<td>TICKVALUESPLITCHARDROP</td>
<td>Specifies whether the split characters are included in the displayed tick values.</td>
</tr>
<tr>
<td>TICKVALUESPLITJUSTIFY</td>
<td>Specifies justification of the strings that are inside the tick value block.</td>
</tr>
</tbody>
</table>

**TICKDISPLAYLIST=(string-list)**

specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the TICKVALUELIST= option. The strings map one-to-one positionally with the values that are listed in the TICKVALUELIST= option.

**Default**
Determined by the system or by the TICKVALUELIST= option.

**Requirements**
The list of values must be enclosed in parentheses.

Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.

**Tip**
This option should be used with the TICKVALUELIST= option. The number of items in the list for this option should equal the number of items in the list for the TICKVALUELIST= option.

**Example**
The following example specifies the axis tick values 10, 20, 30, and 40, and the tick display values A, B, C, and D:

tickvaluelist=("10" "20" "30" "40");
tickdisplaylist=("A" "B" "C" "D");
TICKTYPE= M IDPOINT | INBETWEEN
specifies the position of the axis tick marks.

M IDPOINT
places the tick marks at the midpoint value location.

INBETWEEN
places the tick marks half way between adjacent midpoint locations.

Default MIDPOINT
Restriction This option applies to discrete axes only.

Note Starting with SAS 9.4M2, when TICKTYPE=INBETWEEN, the outermost tick marks and grid lines at each end of the axis are not drawn.

TICKVALUEFITPOLICY= policy
specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

EXTRACT
displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE
makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

SPLIT
splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See TICKVALUESPLITCHAR=
SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See     TICKVALUESPLITCHAR=

SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit the available space.

THIN
eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

EXTRACT
display consecutive integers along the axis instead of the actual tick values to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement          The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement          The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.
ROTATETHIN
attempts the ROTATE policy, and then the THIN policy.

SPLIT
splits the tick value at a split character, which is specified by the
TICKVALUESPLITCHAR= option, only when necessary in order to make the
value fit the available space.

A split does not occur at a split character if a split is not needed at that
location. If the value does not contain any of the specified split characters,
then the value is not split. Values that are not split or that do not fit the
available space even after splitting might overlap the adjoining space.

See  TICKVALUESPLITCHAR=

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is
specified by the TICKVALUESPLITCHAR= option.

See  TICKVALUESPLITCHAR=

SPLITROTATE
attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

Note:  This feature applies to SAS 9.4M5 and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

Note:  This feature applies to SAS 9.4M5 and to later releases.

STAGGER
alternates the tick values between two rows.

STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATERotate
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.
TRUNCATETHIN
attempts the TRUNCATE policy, and then the THIN policy.

Defaults
ROTATE for the X and X2 axes
THIN for the Y and Y2 axes

Note
A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=\textit{format}
specifies how to format the values for major tick marks.

Note: This feature applies to SAS 9.4M3 and to later releases.

Restrictions
This option applies only to discrete axes.

Interaction
This option is ignored when the axis tick values are extracted to an axis legend. See \texttt{ICKVALUEFITPOLICY=EXTRACT | EXTRACTALWAYS} on page 1084.

Tip
Use this option when you want to duplicate tick values on an axis.

TICKVALUELIST=\textit{string-list}
specifies the list of tick values that are to be displayed on the axis.

\textit{string-list}
a space-separated list of values, enclosed in parentheses.

You must enclose each value in the list in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

Requirements
The list of values must be enclosed in parentheses.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.

Notes
If the string list contains duplicate values, then the first occurrence of the duplicated value in the list is honored and the remaining instances are ignored.

When the values specified in the list are compared with the actual data values, leading blanks are honored and trailing blanks are ignored.

Tips
You can use this option to subset the axis values or to display the values in a specific order.

You can use this option to display values on the axis that are not contained in the data.
Examples

The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
tickvaluelist=("Sedan" "Sports" "Wagon" "SUV")
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
tickvaluelist=("10" "20" "30" "40")
```

**TICKVALUEROTATION=**DIAGONAL | DIAGONAL2 | VERTICAL

specifies how the tick values are rotated on the X and X2 axes.

**DIAGONAL**

rotates the tick values to a 45-degree diagonal position.

The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

**DIAGONAL2**

rotates the tick values to a –45-degree diagonal position.

The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**VERTICAL**

rotates the labels to a 90-degree vertical position.

The labels are always drawn from bottom to top.

Default: **DIAGONAL**

Restriction: This option is valid for COLUMNAXIS statements only.

Interaction: The TICKVALUEFITPOLICY= option must be set to ROTATE or ROTATEALWAYS for this option to have any effect.

**TICKVALUESPLITCHAR="character-list"**

specifies a list of characters on which the tick values can be split, if needed. When multiple characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the tick value. In that case, all of the specified split characters together are treated as a single split character.

When a tick value collision is detected, the tick value is split at each occurrence of any of the characters in the character list or all. If all of the split characters occur consecutively in the tick value, then they are treated as a single split character. If the tick value does not contain any of the specified characters, then the value is not split.

```
"character-list"
```

one or more characters with no delimiter between each character.

Default: A blank space

Requirements: The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no delimiters. For example, to specify the characters a, b, and c, use the following option:
Interactions

This option is ignored unless option `TICKVALUEFITPOLICY=` is set to `SPLIT`, `SPLITALWAYS`, `SPLITTHIN`, or `SPLITALWAYSTHIN`.

The `TICKVALUEFITPOLICY=` option sets the policy that is used to manage the split behavior of the tick values.

The `TICKVALUESPLITCHARDROP=` option specifies whether the split characters are displayed or dropped from the display.

Notes

When multiple characters are specified, the order of characters in the list is not significant.

The split characters are case sensitive.

Tips

Use the `TICKVALUESPLITJUSTIFY=` option to specify the justification of the strings in the tick value block.

For the X and X2 axis tick values, use the `TICKVALUEVALIGN=` option to specify the vertical alignment of the tick values.

For the Y and Y2 axis tick values, use the `TICKVALUEHALIGN=` option to specify the horizontal alignment of the tick values.

Example

The following example specifies a blank space, a comma, and an underscore as split characters:

```
tickvaluesplitchar=" ,_"
```

**TICKVALUESPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters should be included in the displayed tick values. The split characters are specified by the `TICKVALUESPLITCHAR=` option.

**TRUE**

drops the split characters from the tick value display.

The following figure shows an example in which `TICKVALUESPLITCHARDROP=TRUE` and three-word, asterisk-delimited tick values are split on the asterisk character by using the `SPLITALWAYS` policy.

Notice that the asterisk delimiter is not displayed.

**FALSE**

includes the split characters in the tick value display.

The fit policy determines how the characters are displayed. If the display policy is `SPLIT` or `SPLITTHIN` and `TICKVALUESPLITCHARDROP=FALSE`, then each tick value is split at a split character only where a split is necessary in order to make the value fit the available space. A split might not occur at every split character in the tick value. At each split point, the split character remains as the last character in the current line. The characters that follow the split character, up to and including the split character at the next split
point, are then wrapped to the following line. This process repeats until the entire data tick value is displayed. The following figure shows an example in which TICKVALUESPLITCHARDROP=FALSE and three-word, asterisk-delimited tick values are split on the asterisk character by using the SPLIT policy.

Notice that a split occurs on the first asterisk and not at the second. In this case, a split is not needed at the second asterisk.

If the fit policy is SPLITALWAYS or SPLITALWAYSTHIN, and TICKVALUESPLITCHARDROP=FALSE, then each tick value is split at every instance of a split character in the value regardless of whether a split is actually needed. Each split character remains as the last character in the current line. The characters that follow each split character, up to and including the next split character, are then wrapped to the next line. The following figure shows an example in which TICKVALUESPLITCHARDROP=FALSE and three-word, asterisk-delimited tick values are split on the asterisk character by using the SPLITALWAYS policy.

Notice that a split occurs after each asterisk and each asterisk appears at the end of the line. In this case, three lines are displayed.

Default: TRUE

Interactions: The TICKVALUESPLITCHAR= option specifies the split character or characters.

This option is ignored unless option TICKVALUEFITPOLICY= is set to SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.

See "boolean " on page 1479 for other Boolean values that you can use.

TICKVALUESPLITCHARJUSTIFY=Center | Left | Right

specifies justification of the strings that are inside the tick value block. The justification is relative to an individual tick value’s display area and does not affect the display of tick values that are not split.
Defaults

- CENTER for an X or X2 axis
- RIGHT for a Y axis
- LEFT for a Y2 axis

Interaction

This option is ignored unless option `TICKVALUEFITPOLICY=` is set to SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.

Options for Row and Column Linear Axes Only

The options that are documented in this section can be used with the `LINEAROPTS=` axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Linear Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>Specifies that evenly spaced integer values are used for tick marks.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKCOUNT</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether the minor tick marks are displayed on the axis.</td>
</tr>
<tr>
<td>THRESHOLDMAX</td>
<td>Specifies a bias for including one more tick mark at the maximum end of the axis.</td>
</tr>
<tr>
<td>THRESHOLDMIN</td>
<td>Specifies a bias for including one more tick mark at the minimum end of the axis.</td>
</tr>
</tbody>
</table>
Linear Axis Option | Description
--- | ---
TICKDISPLAYLIST | Specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.
TICKVALUEFITPOLICY | Specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available in the ROWAXIS statement.
TICKVALUEFORMAT | Specifies how to format the values for major tick marks.
TICKVALUELIST | Specifies the order of the tick values for a linear axis as list.
TICKVALUEPRIORITY | Specifies whether an axis tick specification can extend the axis data range.
TICKVALUEROTATION | Specifies how the tick values are rotated on the X and X2 axes.
TICKVALUESEQUENCE | Specifies the tick values for a linear axis by start, end, and increment.
VIEWMAX | Specifies the maximum data value to include in the display.
VIEWMIN | Specifies the minimum data value to include in the display.

**INTEGER=TRUE | FALSE**
specifies that evenly spaced integer values are used for tick marks.

Default | FALSE
--- | ---
Interactions | This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.
This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option.
INTEGER=TRUE is ignored for the X or X2 axis when a histogram plot is the primary plot and BINAXIS=TRUE is specified in the HISTOGRAM or HISTOGRAMPARM statement.

See | “boolean ” on page 1479 for other Boolean values that you can use.

**MINORGRID=TRUE | FALSE**
specifies whether grid lines are displayed at the minor tick marks.
Defaults

FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips

The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See

“boolean” on page 1479 for other Boolean values that you can use.

**MINORGRIDATTRS=**

- **style-element**
- **style-element (line-options)**
- (line-options)

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

![Linear Axis with Minor Grid Lines](image)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults

The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction

This option is ignored when MINORTICKS=FALSE.
When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Use the GRIDATTRS= option to control the appearance of the major grid lines.

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Here is an example that specifies light blue, dotted lines for the minor grid.

```
minorgridattrs=(color=lightblue pattern=dot);
```

**MINORTICKCOUNT=** **positive-integer**

specifies the number of minor ticks that are displayed on the axis.

Four ticks with five intervals in SAS 9.4M1 and earlier releases.

One tick with two intervals starting with SAS 9.4M2.

The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

To display \( n \) intervals between major ticks, use MINORTICKCOUNT=\( n-1 \).

**MINORTICKS=** **TRUE | FALSE**

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

```
TRUE

\[0\] \[50\] \[100\] \[150\] \[200\]

FALSE

\[0\] \[50\] \[100\] \[150\] \[200\]
```

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Use the MINORGRID= option to display grid lines at the minor tick values.

“boolean ” on page 1479 for other Boolean values that you can use.

**THRESHOLDMAX=** **number**

specifies a bias for including one more tick mark at the maximum end of the axis.
### THRESHOLDMIN=number

specifies a bias for including one more tick mark at the minimum end of the axis.

| Default | 0.30 |
| Range   | 0–1  |
| Restriction | This option applies to linear axes only. |
| Interaction | This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used. |
| Tips | If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed. |
|         | Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range. |
|         | Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks. |
|         | For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0. |

#### See

“Adjusting Axis Thresholds” on page 984

### TICKDISPLAYLIST=(string-list)

specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the TICKVALUELIST= option. The strings map one-to-one positionally with the values that are listed in the TICKVALUELIST= option.

| Default | The display of tick values is controlled by the TICKVALUEFORMAT= option. |
**Requirements**  
The list of values must be enclosed in parentheses.

Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.

**Interaction**  
When this option is specified, the `TICKVALUEFORMAT=` option is ignored.

**Tip**  
This option should be used with the `TICKVALUELIST=` option. The number of items in the list for this option should equal the number of items in the list for the `TICKVALUELIST=` option.

**TICKVALUEFITPOLICY=**`policy`

specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

**EXTRACT**

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

**Requirement**  
The `EXTRACT` policy must be used with an `AXISLEGEND` statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*.

**EXTRACTALWAYS**

same as `EXTRACT`, except that the extraction is implemented regardless of whether collision occurs.

**Requirement**  
The `EXTRACTALWAYS` policy must be used with an `AXISLEGEND` statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*.

**NONE**

makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

**SPLIT**

splits the tick value at a split character, which is specified by the `TICKVALUESPLITCHAR=` option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

**See**  
`TICKVALUESPLITCHAR=`
SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is
specified by the TICKVALUESPLITCHAR= option.

See  TICKVALUESPLITCHAR=

SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words
do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit
the available space.

THIN
eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

EXTRACT
display consecutive integers along the axis instead of the actual tick values to
represent those tick values.

In most cases, this policy is implemented if the system estimates that a
collision might occur. If no collision occurs, then the actual tick values are
displayed on the axis in the normal manner.

Requirement  The EXTRACT policy must be used with an AXISLEGEND
statement. For more information, see “Extracting Discrete Axis
Tick Values into a Legend” in SAS Graph Template Language:
User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of
whether collision occurs.

Requirement  The EXTRACTALWAYS policy must be used with an
AXISLEGEND statement. For more information, see “Extracting
Discrete Axis Tick Values into a Legend” in SAS Graph

NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated
to a 45-degree diagonal or a 90-degree vertical position. By default, the
values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated
to a 45-degree diagonal or a 90-degree vertical position. By default, the
values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if
collisions still occur.
ROTATETHIN

attempts the ROTATE policy, and then the THIN policy.

SPLIT

splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See  TICKVALUESPLITCHAR=

SPLITALWAYS

always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See  TICKVALUESPLITCHAR=

SPLITROTATE

attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS

always displays the tick values vertically as stacked letters.

Note:  This feature applies to SAS 9.4M5 and to later releases.

STACKEDALWAYSTHIN

always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

Note:  This feature applies to SAS 9.4M5 and to later releases.

STAGGER

alters the tick values between two rows.

STAGGERROTATE

attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN

attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE

attempts the STAGGER policy, and then the TRUNCATE policy.

THIN

eliminates alternate tick values.

TRUNCATE

shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE

attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER

attempts the TRUNCATE policy, and then the STAGGER policy.
TRUNCATETHIN

attempts the TRUNCATE policy, and then the THIN policy.

Default  THIN

Note   A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=(format-options) | DATA | format

specifies how to format the values for major tick marks.

(format-options)

specifies one or more formatting options for major tick values.

Together, these options provide parameters for determining an optimal format (w.d, Ew., BESTw.) for displaying major tick values.

MAXWIDTH=integer

specifies the maximum width for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default  8

MAXDECIMALS=integer

specifies the maximum number of decimals for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default  6

Note   The MAXWIDTH= option value should be greater than the MAXDECIMALS= option value.

PREFERREDDECIMALS=integer

specifies the number of decimal places that you want to display for most values.

The actual number might vary based on other constraints.

Default  2

EXTRACTSCALE=TRUE | FALSE

specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.

The scale can be a named scale or a scientific-notation scale. The EXTRACTSCALETYPE= option specifies the scale type. The scale that is used is appended to the axis label, as shown in the following example.

Total Sales (millions)

For long axis labels, if the scale does not fit the available space, then the label is truncated, and the scale is appended to the truncated label. Ellipses indicate that the label was truncated, as shown in the following example.

Total Sales for the Fourth Quarter Of ... (millions)

In extreme cases in which the scale does not fit even with truncation, the entire axis is dropped.

Default  FALSE
Restriction  The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale for all locales.

Interactions  The scale type is determined by the EXTRACTSCALETYPE= option.

If the axis label is not displayed, then the EXTRACTSCALE=TRUE option is ignored.

Note  When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

See  “boolean,” page 1479 for other Boolean values that you can use.

EXTRACTSCALETYPE=DEFAULT | SCIENTIFIC specifies whether to extract a named scale or a scientific-notation scale.

DEFAULT extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as 10^n) for values over 999 trillion. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 (10^-n) for values less than 1 trillionth.

SCIENTIFIC extracts a scientific-notation scale.

A scientific-notation scale is a multiple of 10 expressed as 10^n for values greater than 1, or a multiple of 1/10 expressed as 10^-n for values less than 1.

Default  DEFAULT

Restriction  The scale is derived from the English locale for all locales.

DATA uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

format  specifies a format to apply to the major tick values.

Restriction  GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” page 1541.

Note  If you specify a format that significantly reduces precision, because of tick-value rounding, the plot data elements might not
align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

<table>
<thead>
<tr>
<th>Default</th>
<th>(MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2, EXTRACTSCALE=FALSE, EXTRACTSCALETYPE=DEFAULT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>This option is ignored when the <strong>TICKDISPLAYLIST=</strong> option is specified.</td>
</tr>
</tbody>
</table>

**TICKVALUELIST=** *(numeric-list)*  
specifies the tick values for a linear axis as a list.

<table>
<thead>
<tr>
<th>Default</th>
<th>An internal algorithm determines the tick marks, based on the actual axis data range or the data range established by the <strong>VIEWMIN</strong>= and <strong>VIEWMAX</strong>= options. By default, when this option is used, the only tick values that appear are the tick values in <em>numeric-list</em> that fall within the explicit data range (set by <strong>VIEWMIN</strong>= and <strong>VIEWMAX</strong>=) or the implicit data range (set by the actual data minimum and data maximum).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>This option applies to linear axes only.</td>
</tr>
<tr>
<td>Requirement</td>
<td>The tick values must be specified as a space-separated list of numeric values, enclosed in parentheses.</td>
</tr>
<tr>
<td>Interactions</td>
<td>This option overrides the <strong>INTEGER</strong>= option.</td>
</tr>
</tbody>
</table>

This option is ignored if the **TICKVALUESEQUENCE**= option is specified, or if the **DISPLAY**= option or the **DISPLAYSECONDARY**= option does not display tick values.

The **VIEWMIN**= and **VIEWMAX**= options alter the axis data range. If the **VIEWMIN**= option is set to the minimum tick list value and the **VIEWMAX**= option is set to the maximum tick list value, then all ticks in the tick list are displayed. This might result in some data not being displayed. For example, data might not be displayed when the **VIEWMIN**= value is greater than the actual data minimum, or when the **VIEWMAX**= value is less than actual data maximum.

If **TICKVALUEPRIORITY**= TRUE, then the **VIEWMIN**= and **VIEWMAX**= options are ignored if they are fully enclosed by the *numeric-list*. The tick *numeric-list* can extend the implicit data range of the axis, but cannot reduce it.

If **MINORTICKS**=TRUE is specified and **TICKVALUELIST**= specifies nonuniform tick values such as 3, 5, 11, and 23, then **MINORTICK**=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the **TICKVALUELIST**= option.

| Tip | The values in the list are formatted according to the setting for the **TICKVALUEFORMAT**= option. |

**TICKVALUEPRIORITY**=**TRUE** | **FALSE**  
specifies whether an axis tick specification (**TICKVALUELIST**= or **TICKVALUESEQUENCE**=) can extend the axis data range.
TRUE  
extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

FALSE  
displays only the tick values that are specified by the TICKVALUELIST= option that fall within the explicit data range set by the VIEWMIN= and VIEWMAX= options or by the implicit data range set by the actual data minimum and data maximum.

Default  FALSE

Restriction  This option applies to linear axes only.

Interactions  When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored.

This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is not specified.

This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the tick values.

Note  If the minimum and maximum of the specified values are within the data range, then this option has no effect.

See  "boolean " on page 1479 for other Boolean values that you can use.

TICKVALUEROTATION=DIAGONAL | DIAGONAL2 | VERTICAL 
specifies how the tick values are rotated on the X and X2 axes.

DIAGONAL  
rotates the tick values to a 45-degree diagonal position.

The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

DIAGONAL2  
rotates the tick values to a –45-degree diagonal position.

The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

Note:  This feature applies to SAS 9.4M5 and to later releases.

VERTICAL  
rotates the labels to a 90-degree vertical position.

The labels are always drawn from bottom to top.

Default  DIAGONAL

Restriction  This option is valid for COLUMNAXIS statements only.

Interaction  The TICKVALUEFITPOLICY= option must be set to ROTATE or ROTATEALWAYS for this option to have any effect.
TICKVALUESEQUENCE=(START=number END=number INCREMENT=number) specifies the tick values by start, end, and increment.

**START=number**
- specifies the value for the first tick mark.

**END=number**
- specifies the value for the last tick mark.

**INCREMENT=number**
- specifies the increment for intermediate tick marks between the first and last tick marks.

The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

**Default**
An internal algorithm determines the tick marks, based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default, when this option is used, the only tick values that appear are those that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or the implicit data range (set by the actual data minimum and data maximum).

**Interactions**
- This option overrides the INTEGER= option.
- The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the START= option value and the VIEWMAX= option is set to the END= option value, then all ticks in the tick sequence are displayed.
- If TICKVALUEPRIORITy= TRUE, then the tick sequence might extend the explicit data range of the axis, but never reduce it.
- This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display tick marks.

**Tip**
The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

**See**
- TICKVALUELIST= option as an alternative for customizing tick marks.

**VIEWMAX=number**
specifies the maximum data value to include in the display. The value might be adjusted by the threshold calculation.

**Default**
The maximum value in the data for the specified axis.

**Interactions**
- This option does not determine the maximum axis tick value that is displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.
- This option is ignored when TICKVALUEPRIORITy= TRUE.

**Notes**
- Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

- The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional
factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip
To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

See “Adjusting the Axis View” on page 983

VIEWMIN=number
specifies the minimum data value to include in the display. The value might be adjusted by the threshold calculation.

Default The minimum value in the data for the specified axis.

Interactions This option does not determine the minimum axis tick value that is displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

This option is ignored when TICKVALUEPRIORITY= TRUE.

Notes Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

See “Adjusting the Axis View” on page 983

Options for Row and Column Log Axes Only
The options that are described in this section can be used with the LOGOPTS= axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Log Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASE</td>
<td>Specifies the base of the logarithmic scale for the axis values.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick marks.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKCOUNT</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>TICKINTERVALSTYLE</td>
<td>Specifies how to scale and format the values for major tick marks.</td>
</tr>
</tbody>
</table>
Log Axis Option | Description
--- | ---
TICKVALUEFORMAT= | Specifies how to format the values for major tick marks.
TICKVALUELIST | Specifies the tick values for a log axis as a space-separated list.
TICKVALUEPRIORITY | Specifies whether the TICKVALUELIST specification can extend the axis data range.
VALUESTYPE | Specifies the scale that the system uses when interpreting the TICKVALUELIST=, VIEWMAX=, and VIEWMIN= option values.
VIEWMAX | Specifies the maximum data value to include in the display.
VIEWMIN | Specifies the minimum data value to include in the display.

**BASE=10 | 2 | E**
specifies the base of the logarithmic scale for the axis values.

Default 10
Restriction This option applies to log axes only.

**MINORGRID=TRUE | FALSE**
specifies whether grid lines are displayed at the minor tick marks.

TRUE

FALSE

Defaults FALSE in SAS 9.4M1 and earlier releases.
The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction This option is ignored if the GRIDDISPLAY= option does not display the grid lines.
The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See “boolean” on page 1479 for other Boolean values that you can use.

MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults The GraphGridLines style element is used starting with SAS 9.4.
The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction This option is ignored when MINORTICKS=FALSE.

Note When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip Use the GRIDATTRS= option to control the appearance of the major grid lines.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example Here is an example that specifies light blue, dotted lines for the minor grid.

minorgridattrs=(color=lightblue pattern=dot);

MINORTICKCOUNT=positive-integer
specifies the number of minor ticks that are displayed on the axis.

Default Eight ticks with nine intervals (BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT).
Restriction

Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

Interactions

The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

Tip

To display \( n \) intervals between major ticks, use MINORTICKCOUNT=\( n-1 \).

MINORTICKS=TRUE | FALSE

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

```
TRUE

1  10  100  1000

FALSE

1  10  100  1000
```

Default

FALSE

Restriction

Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

Requirement

If you set this option to TRUE, you must also specify a value for MINORTICKCOUNT=. Otherwise, minor tick marks do not appear on the log axis.

Interaction

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform log tick values such as 10, 20, 30, and 40, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform log tick values such as 1, 10, 100, and 1000 in the TICKVALUELIST= option.

Tip

Use the MINORGRID= option to display grid lines at the minor tick values.

See

“boolean ” on page 1479 for other Boolean values that you can use.

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR

specifies how to scale and format the values for major tick marks.

AUTO

selects a LOGEXPAND, LOGEXPONENT, or LINEAR representation automatically based on the range of the data.

When the data range is small (within an order of magnitude), a LINEAR representation is typically used. Data ranges that encompass several orders of magnitude typically use the LOGEXPAND or LOGEXPONENT representation.

LOGEXPAND

places the major tick marks at uniform intervals at integer powers of the base.

The tick values are expanded as follows:
LOGEXPONENT
places the major tick marks at uniform intervals at integer powers of the base.
The tick values are only the integer exponents for all bases.

LINEAR
places the major tick marks at non-uniform intervals that cover the range of the data.

Default: AUTO
Restrictions: This option applies to log axes only.

For LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LOGEXPAND, formats on data columns contributing to the axis are ignored, although any "named format" on the column is retained. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of w.d, Ew., or BESTw. Other formats (SAS defined or user-defined) are used if specified.

GTL currently honors most but not every SAS format. For details, see Appendix 5, "SAS Formats Not Supported," on page 1541.

Note: When BASE=10 and LOGEXPAND or LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

Tip: If you use TICKINTERVALSTYLE=LOGEXPONENT, then you might want to include information in the axis label about which base is used.

TICKVALUEFORMAT=DATA | format
specifies how to format the values for major tick marks.

Note: This feature applies to SAS 9.4M3 and to later releases.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.
**format**

specifies a format to apply to the major tick values.

**Restriction**

GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

**Restriction**

This option applies to log axes only.

**Interactions**

This option is ignored when `TICKINTERVALSTYLE=LOGEXPONENT`.

When `TICKINTERVALSTYLE=LOGEXPAND`, this option is honored for the base 10 and base 2 logarithmic scales, and is ignored for the base E scale.

When `TICKINTERVALSTYLE=LINEAR`, this option is honored for the base 10, base 2, and base E logarithmic scales.

**See**

`BASE=`

`TICKINTERVALSTYLE=`

**TICKVALUELIST=(numeric-list)**

specifies the tick values for a linear axis as a list.

**Default**

Only the tick values specified in the list that fall within the explicit data range set by the `VIEWMIN=` and `VIEWMAX=` options or by the implicit data range set by the actual data minimum and data maximum are displayed. An internal algorithm determines the tick marks.

**Requirements**

The tick values must be specified as a space-separated list of numeric values, enclosed in parentheses.

The values that you specify must be appropriate for the `VALUESTYPE=` specification. Otherwise, unexpected results might occur. If `VALUESTYPE=EXPANDED` is in effect (default), specify increments of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If `VALUESTYPE=EXPONENT` is in effect, specify integer increments of the log base power exponent such as 1, 2, 3, and so on.

**Interactions**

The `VALUESTYPE=` option determines how the values in the list are interpreted.

The `VIEWMIN=` and `VIEWMAX=` options alter the axis data range. If the `VIEWMIN=` option is set to the minimum tick list value and the `VIEWMAX=` option is set to the maximum tick list value, then all ticks in the tick list are displayed. This might result in some data not being displayed. For example, data might not be displayed when the `VIEWMIN=` value is greater than the actual data minimum, or when the `VIEWMAX=` value is less than actual data maximum.
If the VIEWMIN= value is greater than the actual data minimum or the VIEWMAX= value is less than actual data maximum, some data might not be displayed.

This option is ignored if the DISPLAY= or the DISPLAYSECONDARY= option does not display the tick values.

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform log tick values such as 10, 20, 30, and 40, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform log tick values such as 1, 10, 100, and 1000 in the TICKVALUELIST= option.

See

TICKINTERVALSTYLE= for specifying the scale and format of the major tick values

TICKVALUEPRIORITY= for controlling the behavior of the TICKVALUELIST= option

BASE= for specifying the log base

**TICKVALUEPRIORITY=TRUE | FALSE**

specifies whether the TICKVALUELIST= specification can extend the axis data range.

**TRUE**

extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by the TICKVALUELIST= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

**FALSE**

displays only the tick values that are specified by the TICKVALUELIST= option that fall within the explicit data range set by the VIEWMIN= and VIEWMAX= options or by the implicit data range set by the actual data minimum and data maximum.

Default  

**FALSE**

Interactions

When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored.

This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the tick values.

This option is ignored if the TICKVALUELIST= option is not specified.

**Note**

If the minimum and maximum of the specified values are within the data range, then this option has no effect.

**See**

"boolean " on page 1479 for other Boolean values that you can use.

**VALUESTYPE=EXPANDED | EXPONENT**

specifies the scale that the system uses when interpreting the TICKVALUELIST=, VIEWMAX=, and VIEWMIN= option values. Use this option to choose your preferred way of specifying log-axis values.
EXPANDED
values are interpreted as powers of the base such as 0.1, 1, 10, 100, and so on, for base 10, for example.

EXPONENT
values are interpreted as integer exponents of the base such as 1, 2, 3, and so on, for base 10, base 2, and base E.

Default EXPANDED

Note This option does not change the style of the tick values that are displayed on the axis. It changes only how the VIEWMIN=, VIEWMAX=, and TICKVALUELIST= option values are interpreted by the system.

Tip This option is particularly useful when BASE=E.

Examples The following example specifies VIEWMIN= and VIEWMAX= as exponent values instead of as expanded values on an expanded Base 10 log axis. This results in X-axis tick values of 10, 100, 1000, 10000, and 100000.

```plaintext
xaxisopts=(type=log
    logopts=(base=10
tickintervalstyle=logexpand
valuestype=exponent
viewmin=1 viewmax=5));
```

The following example specifies TICKVALUELIST= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```plaintext
xaxisopts=(type=log
    logopts=(base=10
tickintervalstyle=logexponent
tickvaluepriority=true
valuestype=expanded
tickvaluelist=(10 100 1000 10000 100000));
```

VIEWMAX=number
specifies the maximum data value to include in the display.

Default The maximum data value in the data for the specified axis.

Requirement The value that you specify must be appropriate for the VALUESTYPE= specification and the log base. Otherwise, unexpected results might occur. If VALUESTYPE=EXPANDED is in effect (default), specify an increment of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If VALUESTYPE=EXPONENT is in effect, specify an integer increment of the log base power exponent such as 1, 2, 3, and so on.

Interactions This option is ignored when TICKVALUEPRIORITY= TRUE.

The VALUESTYPE= option determines how the value is interpreted.

Notes Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.
If an invalid value is specified for the VIEWMAX= option, the default value for VIEWMAX= is used instead. In that case, if the default value for VIEWMAX= is less than the value specified by the VIEWMIN= option, then the VIEWMIN= and VIEWMAX= values are swapped.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

When BASE=10 and TICKINTERVALSTYLE=LOGEXPAND or TICKINTERVALSTYLE=LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

**Tip**
To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

**See**
“Adjusting the Axis View” on page 983

**Examples**
The following example specifies a value of 100,000 as an expanded value on a base 10 log axis:

```
VIEWMAX=100000
```

The following example specifies a value of 100,000 as an exponent value on a base 10 log axis:

```
VIEWMAX=5
```

**VIEWMIN=number**
specifies the minimum data value to include in the display.

**Default**
The minimum value in the data for the specified axis.

**Requirement**
The value that you specify must be appropriate for the VALUESTYPE= specification and the log base. Otherwise, unexpected results might occur. If VALUESTYPE=EXPANDED is in effect (default), specify an increment of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If VALUESTYPE=EXPONENT is in effect, specify an integer increment of the log base power exponent such as 1, 2, 3, and so on.

**Interactions**
This option is ignored when TICKVALUEPRIORITY= TRUE.

**Notes**
Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

When BASE=10 and TICKINTERVALSTYLE=LOGEXPAND or TICKINTERVALSTYLE=LOGEXPONENT is used, an intermediate
tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

Tip
To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

See
“Adjusting the Axis View” on page 983

Examples
The following example specifies a value of 0.1 as an expanded value on a base 10 log axis:
VIEWMIN=0.1

The following example specifies a value of 0.1 as an exponent value on a base 10 log axis:
VIEWMIN=-1

Options for Row and Column Time Axes Only
The options that are documented in this section can be used with the TIMEOPTS= axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Time Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
<td>Specifies the time interval between major tick marks.</td>
</tr>
<tr>
<td>INTERVALMULTIPLIER</td>
<td>Specifies a multiplier to apply to the time interval that is in effect for the axis.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKINTERVAL</td>
<td>Specifies the time interval between minor ticks.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor tick marks are displayed.</td>
</tr>
<tr>
<td>SPLITTICKVALUE</td>
<td>Specifies whether to split the tick values on column axes, if possible. This option is not available in the ROWAXIS statement.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision on column axes. This option is not available in the ROWAXIS statement.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values for a time axis as list.</td>
</tr>
<tr>
<td>Time Axis Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification can extend the axis data range.</td>
</tr>
<tr>
<td>TICKVALUEROTATION</td>
<td>Specifies how the tick values are rotated on the X and X2 axes.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display.</td>
</tr>
</tbody>
</table>

**INTERVAL=** AUTO | SECOND | MINUTE | HOUR | DAY | TENDAY | WEEK | SEMIMONTH | MONTH | QUARTER | SEMIYEAR | YEAR  

specifies the time interval between major ticks.

**Table 8.3 Time Intervals**

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>DATE, TIME, or DATETIME</td>
<td>automatically chosen</td>
<td>automatically chosen</td>
</tr>
<tr>
<td>SECOND</td>
<td>TIME or DATETIME</td>
<td>second</td>
<td>TIME8.</td>
</tr>
<tr>
<td>MINUTE</td>
<td>TIME or DATETIME</td>
<td>minute</td>
<td>TIME8.</td>
</tr>
<tr>
<td>HOUR</td>
<td>TIME or DATETIME</td>
<td>hour</td>
<td>TIME8.</td>
</tr>
<tr>
<td>DAY</td>
<td>DATE or DATETIME</td>
<td>day</td>
<td>DATE9.</td>
</tr>
<tr>
<td>TENDAY</td>
<td>DATE or DATETIME</td>
<td>10 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>WEEK</td>
<td>DATE or DATETIME</td>
<td>7 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>SEMIMONTH</td>
<td>DATE or DATETIME</td>
<td>1st and 16th of each month</td>
<td>DATE9.</td>
</tr>
<tr>
<td>MONTH</td>
<td>DATE or DATETIME</td>
<td>month</td>
<td>MONYY7.</td>
</tr>
<tr>
<td>QUARTER</td>
<td>DATE or DATETIME</td>
<td>3 months</td>
<td>YYQC6.</td>
</tr>
</tbody>
</table>
INTERVAL= unit

Tick interval

Default tick value

format

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value</th>
<th>format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEMIYEAR</td>
<td>DATE or</td>
<td>6 months</td>
<td>MONYY7.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DATETIME</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YEAR</td>
<td>DATE or</td>
<td>year</td>
<td>YEAR4.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DATETIME</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default: AUTO. An appropriate interval is chosen based on the data and the column date, date-time, or time format.

Restriction: This option applies to time axes only.

Requirement: The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an interval must be consistent with the duration unit. For example, if the data are in time units, you can specify only AUTO, SECOND, MINUTE, HOUR.

Interaction: This option is ignored if the TICKVALUELIST= option is used.

INTERVALMULTIPLIER= positive-integer

specifies a multiplier to apply to the time interval that is in effect for the axis.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default: 1

Restriction: This option applies to time axes only.

Interaction: This option is ignored if the TICKVALUELIST= option is used.

Tip: Use the INTERVAL= option to specify a different time interval.

Examples: To specify 3-month intervals:

INTERVAL=MONTH INTERVALMULTIPLIER=3

To specify 10-year intervals:

INTERVAL=YEAR INTERVALMULTIPLIER=10

MINORGRID= TRUE | FALSE

specifies whether grid lines are displayed at the minor tick marks.
The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction
This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips
The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See “boolean ” on page 1479 for other Boolean values that you can use.

MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults
The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction
This option is ignored when MINORTICKS=FALSE.
### MINORTICKINTERVAL="interval"

Specifies the time interval between minor ticks. See Table 8.10 on page 1114 for information about the intervals that you can select. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, then you must specify AUTO, SECOND, MINUTE, or HOUR.

**Default**: AUTO

**Interactions**: This option is ignored if the TICKVALUELIST= option is used. This option is ignored if the MINORTICKINTERVAL= setting is greater than the INTERVAL= setting.

### MINORTICKS=TRUE | FALSE

Specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

![Minor Ticks](image)

**Default**: FALSE

**Interactions**: The number of minor ticks is dependent on the value of the MINORTICKINTERVAL= option, if specified. If MINORTICKINTERVAL= is not specified, then it is dependent on the value of the INTERVAL= option.

This option is ignored if the TICKVALUELIST= option is used, or if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick marks.

If MINORTICKS=TRUE is specified with the TICKVALUELIST= option, then MINORTICKS=TRUE is ignored, and minor tick marks are not displayed.

**Tip**: Use the MINORGRID= option to display grid lines at the minor tick values.
SPLITTickValue=TRUE | FALSE
specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available for a Y or Y2 axis.

TRUE
splits the axis tick values into two lines allowing more tick values to appear.

For example, with INTERVAL= MONTH, this is how tick values are split:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td></td>
<td></td>
<td></td>
<td>2004</td>
<td></td>
<td></td>
<td></td>
<td>2005</td>
<td></td>
<td></td>
<td></td>
<td>2006</td>
</tr>
</tbody>
</table>

FALSE
does not split the axis tick values.

For example, when this option specifies FALSE, this is how the tick values in the previous example appear:

|---------|---------|---------|---------|---------|---------|---------|

Typically, fewer tick values fit, causing thinning, rotation, or staggering of the values. See the TICKValueFitPolicy= option.

Default TRUE

Restriction This option applies to time axes only.

Interaction This option is ignored if the TICKValueList= or TICKValueFormat= option is used.

See “boolean ” on page 1479 for other Boolean values that you can use.

TICKValueFitPolicy=policy
specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available for the Y and Y2 axes. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. The following policies are valid:

EXTRACT
displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLegend statement. For more information, see "Extracting Discrete Axis Tick Values into a Legend" in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.
Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE makes no attempt to avoid collisions between tick values. Tick values are displayed even when they collide.

SPLIT splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See TICKVALUESPLITCHAR=

SPLITALWAYS always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITALWAYSTHIN same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

SPLITTHIN same as SPLIT, except that thinning is performed when long words do not fit the available space.

THIN eliminates alternate tick values.

EXTRACT display consecutive integers along the axis instead of the actual tick values to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.
NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The \texttt{TICKVALUEROTATION=} option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The \texttt{TICKVALUEROTATION=} option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.

ROTATETHIN
attempts the ROTATE policy, and then the THIN policy.

SPLIT
splits the tick value at a split character, which is specified by the \texttt{TICKVALUESPLITCHAR=} option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

\textbf{See} \texttt{TICKVALUESPLITCHAR=}

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the \texttt{TICKVALUESPLITCHAR=} option.

\textbf{See} \texttt{TICKVALUESPLITCHAR=}

SPLITROTATE
attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

\textbf{Note:} This feature applies to \texttt{SAS 9.4M5} and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

\textbf{Note:} This feature applies to \texttt{SAS 9.4M5} and to later releases.

STAGGER
alternates the tick values between two rows.
STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.

TRUNCATETHIN
attempts the TRUNCATE policy, and then the THIN policy.

Default       THIN
Restriction   This option is valid only for the X and X2 axes.
Interaction   When SPLITTICKVALUE= TRUE, this option is ignored and only the
              THIN policy is used.
Note          A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=\texttt{format} | \texttt{DATA}
specifies how to format the values for major tick marks.

\texttt{format}
specifies a SAS date, time, or datetime format to control how the major tick
values are displayed.

This format must be in the same duration units as the data column(s)
mapped to a time axis: TIME, DATE, or DATETIME and should be
appropriate for the value of the INTERVAL= option. For example, if
INTERVAL=MONTH and there are two years of data displayed on the axis,
then choosing TICKVALUEFORMAT=YEAR. would result in several ticks
having the same year value.

\texttt{DATA}
specifies that the SAS date, time, or datetime format associated with the data
column assigned to the axis be used to control how the major tick values are
displayed.

Default       The default format used by the INTERVAL= option. The default
does not apply if TICKVALUESLIST= is specified.
Restrictions  This option applies to time axes only.

GTL currently honors most but not every SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.
**TICKVALUELIST=(time-constant-list | date-constant-list | datetime-constant-list | numeric-list)**

specifies the tick values for a time axis as list.

**Default**
An internal algorithm determines the tick values.

**Restrictions**
This option applies to time axes only.

If **TICKVALUEPRIORITY=** is set to FALSE, then this option does not extend the data range of the axis. If the values fall within the default data range or that specified by the **VIEWMIN=** or **VIEWMAX=** options, then they are used.

**Requirement**
The tick values must be specified as a space-separated list of values enclosed in parentheses. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

**Interactions**
The values in the list are formatted according to the format specified on the **TICKVALUEFORMAT=** option. If **TICKVALUEFORMAT=** is not used, then the values are formatted according to the column format (the default **TICKVALUEFORMAT** value is not applied to these values).

If this option is specified, the **SPLITTICKVALUE=** and **INTERVAL=** options are ignored.

If **MINORTICKS=TRUE** is specified with the **TICKVALUELIST=** option, **MINORTICKS=TRUE** is ignored, and minor tick marks are not displayed.

**TICKVALUEPRIORITY=**TRUE | FALSE

specifies whether the **TICKVALUELIST=** specification can extend the axis data range.

**TRUE**
extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by the **TICKVALUELIST=** option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

**FALSE**
displays only the tick values that are specified by the **TICKVALUELIST=** option that fall within the explicit data range set by the **VIEWMIN=** and **VIEWMAX=** options or by the implicit data range set by the actual data minimum and data maximum.

**Default**
**FALSE**

**Interactions**
When this option is set to **TRUE**, the **VIEWMIN=** and **VIEWMAX=** options are ignored.

This option is ignored if the **DISPLAY=** option or the **DISPLAYSECONDARY=** option does not display the tick values.
This option is ignored if the TICKVALUELIST= option is not specified.

Note  If the minimum and maximum of the specified values are within the data range, then this option has no effect.

See  “boolean” on page 1479 for other Boolean values that you can use.

TICKVALUEROTATION=DIAGONAL | DIAGONAL2 | VERTICAL
specifies how the tick values are rotated on the X and X2 axes.

DIAGONAL
rotates the tick values to a 45-degree diagonal position.
The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

DIAGONAL2
rotates the tick values to a –45-degree diagonal position.
The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

Note: This feature applies to SAS 9.4M5 and to later releases.

VERTICAL
rotates the labels to a 90-degree vertical position.
The labels are always drawn from bottom to top.

Default  DIAGONAL

Restriction  This option is valid for COLUMNAXIS statements only.

Interaction  The TICKVALUEFITPOLICY= option must be set to ROTATE or ROTATEALWAYS for this option to have any effect.

VIEWMAX=number
specifies the maximum data value to include in the display.

Default  The maximum value in the data for the specified axis.

Notes  Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip  To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

See  “Adjusting the Axis View” on page 983

VIEWMIN=number
specifies the minimum data value to include in the display.

Default  The minimum value in the data for the specified axis.
Notes Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

See “Adjusting the Axis View” on page 983

Details

The LAYOUT LATTICE statement creates a grid of graphs that automatically aligns plot areas, data display areas, axis labels, and headers across the columns and rows of the layout. The axis data ranges can be scaled, and the axes for individual cells in the layout can be managed by row and by column using COLUMNAXIS and ROWAXIS statements. COLUMNAXIS statements are used within a COLUMNAXES or COLUMN2AXES block to externalize column axes for the layout. Similarly, ROWAXIS statements are used within a ROWAXES or ROW2AXES block to externalize row axes for the layout. Each axis block is used to manage the primary axis. The axis that is considered primary depends on the settings for the XAXIS= and YAXIS= options in plot statements that are specified within the layout:

<table>
<thead>
<tr>
<th>Option Setting</th>
<th>Primary Axis</th>
<th>Axis Block to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAXIS=X</td>
<td>X (bottom)</td>
<td>COLUMNAXES</td>
</tr>
<tr>
<td>XAXIS=X2</td>
<td>X2 (top)</td>
<td>COLUMN2AXES</td>
</tr>
<tr>
<td>YAXIS=Y</td>
<td>Y (left)</td>
<td>ROWAXES</td>
</tr>
<tr>
<td>YAXIS=Y2</td>
<td>Y2 (right)</td>
<td>ROW2AXES</td>
</tr>
</tbody>
</table>

For the specifications to take effect,

- UNION or UNIONALL data scaling must be set for the affected columns and rows. The data scaling is set with the LAYOUT LATTICE statement’s COLUMNDATARANGE=, COLUMN2DATARANGE=, ROWDATARANGE=, and ROW2DATARANGE= options.

- Within a COLUMNAXES or COLUMN2AXES block, one COLUMNAXIS statement should be specified for each column that contains axes that you need to manage. Both axes blocks can contain a COLUMNAXIS statement for the same column. For example, to manage the axes in the first column of the layout, the COLUMNAXES block can contain a COLUMNAXIS statement that manages the column’s X axes. The COLUMN2AXES block can contain a COLUMNAXIS statement that manages the column’s X2 axes.

- Within a ROWAXES or ROW2AXES block, one ROWAXIS statement should be specified for each row that contains axes that you need to manage. Both axes blocks can contain a ROWAXIS statement for the same row. For example, to manage the axes in the first row of the layout, the ROWAXES block can contain
a ROWAXIS statement that manages the row’s Y axes. The ROW2AXES block can contain a ROWAXIS statement that manages the column’s Y2 axes.

In addition to managing the primary axes, you can also display "secondary" axes in the grid. A secondary axis is not an independent axis. Rather, it mirrors the primary axis, but it is displayed on the opposite side and can have different display options. For example, when the X axis (bottom) is primary, you can mirror that axis with a secondary X axis at the top of the grid. Similarly, when the Y2 axis (right) is primary, you can mirror that axis with a secondary Y2 axis on the left of the grid. A secondary axis makes it easier to interpolate values in the cells that are farthest away from the primary axis. To display a secondary axis, use the DISPLAYSECONDARY= option.

For general information about managing primary and secondary axes, see “Plot Data Are Mapped to a Designated Axis” on page 974. For details about managing the axes within a LAYOUT LATTICE, see the discussion about the LAYOUT LATTICE “Row and Column Axis Statements” on page 136.

The following example shows COLUMNAXIS statements for a lattice with two columns:

```
layout lattice / columns=2 columndatarange=union;
    columnaxes;
        columnaxis / griddisplay=on displaysecondary=(ticks tickvalues);
        columnaxis / griddisplay=on displaysecondary=(ticks tickvalues);
    endcolumnaxes;
/* rest of lattice definition */
endlayout;
```

COLUMNAXIS and ROWAXIS statements are similar to the XAXISOPTS= and YAXISOPTS= options for LAYOUT OVERLAY, with the following differences:

- When COLUMNAXIS and ROWAXIS are used, any axis options specified on plots within the affected columns or rows are ignored. All axis features for the row and column axes must be specified on the ROWAXIS or COLUMNAXIS statement.
- When ROWAXIS or COLUMNAXIS are used, any LAYOUT OVERLAYEQUATED layouts specified for cells in the affected columns or rows are implemented as LAYOUT OVERLAY layouts. Equated axes are not supported on row and column axes.

In the default cases for each plot in the layout, the axis type is always DISCRETE, LINEAR, or TIME. The TYPE= option enables you to specify an axis type that overrides the default. For example, when appropriate for the data, you can request a LOG axis. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.

Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types: DISCRETEOPTS=, LINEAROPTS=, LOGOPTS=, and TIMEOPTS=. One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

Example: Row and Column Axis Options for LAYOUT LATTICE

This example shows how to externalize axes in a LAYOUT LATTICE and manage the axis features on primary Y and Y2 axes.
The first HISTOGRAM statement specifies YAXIS=Y2 to make the Y2 axis the primary axis for COUNT measures. The second HISTOGRAM statement specifies YAXIS=Y to make the Y axis the primary axis for PERCENT measures.

In order to externalize the axes within the layout, the data ranges for the axes must be unified. In the LAYOUT LATTICE statement, the ROWDATARANGE= option unifies the data ranges for the Y axes across the row. The ROW2DATARANGE= option unifies the data ranges for the Y2 axes across the row.

The ROWAXIS statement is used to manage axis features for the row axes. To manage the primary Y axis, a ROWAXIS statement is specified within a ROWAXES block. To manage the primary Y2 axis, another ROWAXIS statement is specified within a ROW2AXES block.

Within the ROWAXES block, the ROWAXIS statement consolidates Y axes in the row into a single, column Y axis and also displays grid lines. Within the ROW2AXES block, the ROWAXIS statement consolidates Y2 axes in the row into a single row Y2 axis, but it does not alter the default features of that axis.

The following graph was generated by the “Example Program” on page 1126:

```
proc template;
   define statgraph y2axis;
      begingraph;
         layout lattice / columns=2 columngutter=10
            rowdatarange=union row2datarange=union ;
            rowaxes;
            rowaxis / griddisplay=on;
      endrowaxes;
      row2axes;
```
Axis Options for LAYOUT OVERLAYEQUATED

Axis options for the plots within an OVERLAYEQUATED layout.

Interaction:  The OVERLAYEQUATED’s axis options are ignored when the LAYOUT OVERLAYEQUATED statement is nested within another layout type that has external axes in effect. For example, the axis options are ignored when the statement is nested within a LAYOUT LATTICE with a COLUMNAXIS= or ROWAXIS= option in effect.

See:  “LAYOUT OVERLAYEQUATED Statement” on page 157

Syntax

Axis options for the plots within an OVERLAYEQUATED layout are specified with the following options on a LAYOUT OVERLAYEQUATED statement:

- **COMMONAXISOPTS=**(common-equated-axis-options)
- **XAXISOPTS=**(equated-axis-options)
- **YAXISOPTS=**(equated-axis-options)

Options That Apply to Both Equated Axes

The options that are documented in this section are specified with the COMMONAXISOPTS= option and are applied to both the X and Y axes. With the exception of VIEWMAX and VIEWMIN, these options cannot be applied separately to an X or Y axis using the XAXISOPTS= or YAXISOPTS= option. See “Options That Apply Separately to an X or Y Equated Axis” for a list of options that can be applied to a single axis. The following table provides a summary of the common options.
### Equated Axis Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>Specifies that evenly spaced integer values are used for tick marks for all axes.</td>
</tr>
<tr>
<td>LINEEXTENT</td>
<td>Specifies the extent of the axis lines.</td>
</tr>
<tr>
<td>TICKSTYLE</td>
<td>Specifies the placement of tick marks in relation to the axis line.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values as list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification can extend the axis data range.</td>
</tr>
<tr>
<td>TICKVALUESEQUENCE</td>
<td>Specifies the tick values by start, end, and increment.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display on the X and Y axes when the axis lengths and major tick values are equal.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display on the X and Y axes when the axis lengths and major tick values are equal.</td>
</tr>
</tbody>
</table>

#### INTEGER=TRUE | FALSE

specifies that evenly spaced integer values are used for tick marks.

- **Default**: FALSE
- **Interactions**: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.
  - This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option.
  - INTEGER=TRUE is ignored for the X or X2 axis when a histogram plot is the primary plot and BINAXIS=TRUE is specified in the HISTOGRAM or HISTOGRAMPARM statement.

- **See**: "boolean " on page 1479 for other Boolean values that you can use.

#### LINEEXTENT=FULL | DATA | number

specifies the extent of the axis lines.

- **Note**: This feature applies to SAS 9.4M3 and to later releases.
  - **FULL**: specifies axis lines that extend along the entire length of the axis.
DATA specifies axis lines that extend through the data range from the minimum offset to the maximum offset.

number specifies how much the axis lines extend from DATA toward FULL as a decimal proportion. A value of 0 is equivalent to DATA, and a value of 1 is equivalent to FULL.

Range 0–1

Tip A numeric value is useful for bar charts when DATA terminates the axis line at the midpoint positions of the minimum and maximum bars. In that case, you can specify a numeric value to lengthen the axis line so that it extends to the full width of both bars.

The following figure shows a simple example of each value for the X axis and Y-axis lines. The light-blue dashed lines depict the minimum and maximum offsets that are set on the axes.

```
<table>
<thead>
<tr>
<th>Full</th>
<th>Data</th>
<th>0.5</th>
</tr>
</thead>
</table>
```

Default FULL

Restriction This option is valid only in OVERLAY and OVERLAYEQUATED layouts.

Interaction This option overrides the AXISLINEEXTENT= option in the BEGINGRAPH statement.

Tip The graph wall outline might appear to be an axis line. In that case, use the WALLDISPLAY=NONE or WALLDISPLAY=(FILL) option in the layout statement to suppress the wall outline.

**TICKSTYLE=OUTSIDE | INSIDE | ACROSS**

specifies the placement of tick marks in relation to the axis line. The figure shows the tick display for each value.

```
<table>
<thead>
<tr>
<th>Outside</th>
<th>Inside</th>
<th>Across</th>
</tr>
</thead>
</table>
```

OUTSIDE displays tick marks outside of the axis frame.
**INSIDE**
displays tick marks inside the axis frame.

**ACROSS**
displays tick marks across the axis line.

**Default**

The GraphAxisLines:TickDisplay style reference.

**Interaction**
This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick marks.

**Notes**
This option has no effect on the placement of the tick values, which are always outside the axis frame.

This option applies to both major ticks and minor ticks.

**TICKVALUELIST=(numeric-list)**
specifies the tick values for a linear axis as a list.

**Default**

An internal algorithm determines the tick marks, based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default, when this option is used, the only tick values that appear are the tick values in numeric-list that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or the implicit data range (set by the actual data minimum and data maximum).

**Requirement**
The tick values must be specified as a space-separated list of numeric values, enclosed in parentheses.

**Interactions**
This option overrides the INTEGER= option.

This option is ignored if the LAYOUT OVERLAYEQUATED statement specifies EQUATETYPE=FIT (the default).

This option is ignored if the TICKVALUESEQUENCE= option is specified, or if the DISPLAY= option or the DISPLAYSECONDARY= option does not display tick values.

The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the minimum tick list value and the VIEWMAX= option is set to the maximum tick list value, then all ticks in the tick list are displayed. This might result in some data not being displayed. For example, data might not be displayed when the VIEWMIN= value is greater than the actual data minimum, or when the VIEWMAX= value is less than actual data maximum.

If TICKVALUEPRIORITY= TRUE, then the VIEWMIN= and VIEWMAX= options are ignored if they are fully enclosed by the numeric-list. The tick numeric-list can extend the implicit data range of the axis, but cannot reduce it.

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.
Tip

The values in the list are formatted according to the setting for the TICKVALUEFORMAT= option.

TICKVALUEPRIORITY=TRUE | FALSE

specifies whether an axis tick specification (TICKVALUELIST= or TICKVALUESEQUENCE=) can extend the axis data range.

TRUE

extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by either the TICKVALUELIST= or TICKVALUESEQUENCE= option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

FALSE

displays only the tick values that are specified by the TICKVALUELIST= option that fall within the explicit data range set by the VIEWMIN= and VIEWMAX= options or by the implicit data range set by the actual data minimum and data maximum.

Default

FALSE

Interactions

When this option is set to TRUE, the VIEWMIN= and VIEWMAX= options are ignored.

This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is not specified.

This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the tick values.

Note

If the minimum and maximum of the specified values are within the data range, then this option has no effect.

See

“boolean ” on page 1479 for other Boolean values that you can use.

TICKVALUESEQUENCE=(START=number END=number INCREMENT=number)

specifies the tick values by start, end, and increment.

START=number

specifies the value for the first tick mark.

END=number

specifies the value for the last tick mark.

INCREMENT=number

specifies the increment for intermediate tick marks between the first and last tick marks.

The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

Default

An internal algorithm determines the tick marks, based on the actual axis data range or the data range established by the VIEWMIN= and VIEWMAX= options. By default, when this option is used, the only tick values that appear are those that fall within the explicit data range (set by VIEWMIN= and VIEWMAX=) or the implicit data range (set by the actual data minimum and data maximum).
Interactions This option overrides the INTEGER= option.

The VIEWMIN= and VIEWMAX= options alter the axis data range. If the VIEWMIN= option is set to the START= option value and the VIEWMAX= option is set to the END= option value, then all ticks in the tick sequence are displayed.

If TICKVALUEPRIORITY= TRUE, then the tick sequence might extend the explicit data range of the axis, but never reduce it.

This option is ignored if the LAYOUT OVERLAYEQUATED statement specifies EQUATETYPE=FIT (the default), or if the DISPLAY= option or the DISPLAYSECONDARY= option does not display tick marks.

Tip The values in the sequence are formatted according to the setting for the TICKVALUEFORMAT= option.

See TICKVALUELIST= option as an alternative for customizing tick marks.

**VIEWMAX=**<br>
**number**<br>
specifies the maximum data value to include in the display on the X and Y axes when the axis lengths and major tick values are equal. The value might be adjusted by the threshold calculation.

Default The maximum value in the data for the X and Y axes.

Restriction This option is honored only when EQUATETYPE=SQUARE.

Interactions This option does not determine the maximum axis tick value that is displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

This option is ignored when TICKVALUEPRIORITY= TRUE.

Notes Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

See "Adjusting the Axis View" on page 983

**VIEWMIN=**<br>
**number**<br>
specifies the minimum data value to include in the display on the X and Y axes when the axis lengths and major tick values are equal. The value might be adjusted by the threshold calculation.

Default The minimum value in the data for the X and Y axes.

Restriction This option is honored only when EQUATETYPE=SQUARE.
Interactions

This option does not determine the minimum axis tick value that is displayed. The \texttt{THRESHOLDMIN=} value is used to determine the minimum tick value.

This option is ignored when \texttt{TICKVALUEPRIORITY=} \texttt{TRUE}.

Notes

Setting a \texttt{VIEWMAX=} or \texttt{VIEWMIN=} value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the \texttt{VIEWMIN=} value. The \texttt{VIEWMIN=} and \texttt{VIEWMAX=} values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip

To display the \texttt{VIEWMIN=} value as the minimum tick value, use the \texttt{TICKVALUELIST=} option.

See

“Adjusting the Axis View” on page 983

Options That Apply Separately to an X or Y Equated Axis

The options that are documented in this section can be applied to an X axis with the \texttt{XAXISOPTS=} option, or to the Y axis with the \texttt{YAXISOPTS=} option. See “Options That Apply to Both Equated Axes” on page 1127 for a list of options that apply in common to both axes. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Equated Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{DISPLAY}</td>
<td>Controls which axis features are displayed on the primary axis.</td>
</tr>
<tr>
<td>\texttt{DISPLAYSECONDARY}</td>
<td>Controls which axis features are displayed on the secondary axis.</td>
</tr>
<tr>
<td>\texttt{GRIDATTRS}</td>
<td>Specifies the attributes of the grid lines.</td>
</tr>
<tr>
<td>\texttt{GRIDDDISPLAY}</td>
<td>Specifies when axis grid lines are displayed.</td>
</tr>
<tr>
<td>\texttt{LABEL}</td>
<td>Specifies the axis label.</td>
</tr>
<tr>
<td>\texttt{LABELATTRS}</td>
<td>Specifies the color and font attributes of the axis label.</td>
</tr>
<tr>
<td>\texttt{LINEEXTENT}</td>
<td>Specifies the extent of the axis line.</td>
</tr>
<tr>
<td>\texttt{MINORGRID}</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>\texttt{MINORGRIDATTRS}</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>\texttt{MINORTICKCOUNT}</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>Equated Axis Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether the minor tick marks are displayed on the axis.</td>
</tr>
<tr>
<td>OFFSETMAX</td>
<td>Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>OFFSETMIN</td>
<td>Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>REVERSE</td>
<td>Specifies whether tick values should appear in the reverse order.</td>
</tr>
<tr>
<td>SHORTLABEL</td>
<td>Specifies an alternate axis label to use if the default or specified axis label is too long for the axis length.</td>
</tr>
<tr>
<td>THRESHOLDMAX</td>
<td>Specifies a bias for including one more tick mark at the maximum end of the axis.</td>
</tr>
<tr>
<td>THRESHOLDMIN</td>
<td>Specifies a bias for including one more tick mark at the minimum end of the axis.</td>
</tr>
<tr>
<td>TICKVALUEATTRS</td>
<td>Specifies the color and font attributes of the axis tick values.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision. Only the default policy (THIN) is available for a Y or Y2 axis.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for tick marks.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display on the X or Y axis when the axis lengths and major tick values are not equal.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display on the X or Y axis when the axis lengths and major tick values are not equal.</td>
</tr>
</tbody>
</table>

**DISPLAY=**

- STANDARD: specifies that the LABEL, LINE, Ticks, and TICKVALUES are displayed
- ALL: currently the same as STANDARD
- NONE: specifies that no axis features are displayed

This controls which axis features are displayed on the primary axis.
(display-options)
  specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

  LABEL
    displays the axis label

  LINE
    displays the axis line

  TICKS
    displays the tick marks

  TICKVALUES
    displays the values that are represented by the major tick marks

Default  STANDARD

Tips  The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

  Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

  When LINE is excluded from the DISPLAY= option, the layout wall outline or the default baseline of a bar chart, needle plot, or waterfall chart can appear to be an axis line. To suppress the wall outline, use the WALLDISPLAY= option in the layout statement. To suppress the plot baseline, use the BASELINEATTRS= option in the plot statement.

DISPLAYSECONDARY=NONE | ALL | STANDARD | (display-options)
controls which axis features are displayed on the secondary axis. When data are mapped to the X or Y axis, you can display an X2 or Y2 (secondary) axis using this option. The secondary axis is a duplicate of the X or Y axis but can have different display options.

NONE
  specifies that no axis features are displayed

STANDARD
  specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL
  currently the same as STANDARD

(display-options)
  specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

  LABEL
    displays the axis label

  LINE
    displays the axis line

  TICKS
    displays the tick marks
TICKVALUES displays the values that are represented by the major tick marks

Default NONE

Tip Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

GRIDATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the grid lines.

Default The GraphGridLines style element.

Interaction This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF
specifies whether axis grid lines are displayed. This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

AUTO_OFF specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON specifies that grid lines are always displayed.

The current style has no override.

OFF specifies that grid lines are never displayed.

The current style has no override.

The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

<table>
<thead>
<tr>
<th>GRIDDISPLAY= option</th>
<th>DisplayOpts= style attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_OFF</td>
<td>AUTO</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>GRIDDISPLAY= option</td>
<td>DisplayOpts= style attribute</td>
<td>Grid Lines Shown?</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>AUTO</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>ON</td>
<td>any value</td>
<td>yes</td>
</tr>
<tr>
<td>OFF</td>
<td>any value</td>
<td>no</td>
</tr>
</tbody>
</table>

Default: AUTO_OFF

Note: Supplied styles use DisplayOpts="AUTO," which means that the style has no preference about grid lines and the graphics template setting for grid lines is always used.

**LABEL=string | ("string" ... "string")**
specifies the axis label. The string can be either a string literal or a dynamic. The list form implies that all included string literals or dynamic variables will be concatenated.

Default: The default label is derived from the primary plot in the layout. For more information, see “When Plots Share Data and a Common Axis” on page 979.

Note: If the axis label is too long to fit along the axis, then it is truncated by default.

Tips: Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Use the ODS escape sequence to specify Unicode characters in an axis label. Here is an example that species degrees Fahrenheit in an axis label for temperature:

```
label="Temperature (*ESC*){unicode '00B0'x}F"
```

See: “Overriding the Default Axis Label” in SAS Graph Template Language: User’s Guide

**LABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the axis label.

Default: The GraphLabelText style element.

Interaction: This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

See: “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.
LINEEXTENT=FULL | DATA | number

specifies the extent of the axis line.

---

**Note:** This feature applies to SAS 9.4M3 and to later releases.

**FULL**

specifies an axis line that extends along the entire length of the axis.

**DATA**

specifies an axis line that extends through the data range from the minimum offset to the maximum offset.

**number**

specifies how much the axis line extends from DATA toward FULL as a decimal proportion. A value of 0 is equivalent to DATA, and a value of 1 is equivalent to FULL.

**Range** 0–1

**Tip** A numeric value is useful for bar charts when DATA terminates the axis line at the midpoint positions of the minimum and maximum bars. In that case, you can specify a numeric value to lengthen the axis line so that it extends to the full width of both bars.

The following figure shows a simple example of each value for the X axis and Y-axis lines. The light-blue dashed lines depict the minimum and maximum offsets that are set on the axes.

---

**Default** FULL

**Restriction** This option is valid only in OVERLAY and OVERLAYEQUATED layouts.

**Interactions** This option overrides the AXISLINEEXTENT= option in the BEGINGRAPH statement.

This option overrides the LINEEXTENT= option in COMMONAXISOPTS=.

**Tip** The graph wall outline might appear to be an axis line. In that case, use the WALLDISPLAY=NONE or WALLDISPLAY=(FILL) option in the layout statement to suppress the wall outline.

---

MINORGRID=TRUE | FALSE

specifies whether grid lines are displayed at the minor tick marks.
Defaults FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.

Interaction This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See “boolean ” on page 1479 for other Boolean values that you can use.

MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction This option is ignored when MINORTICKS=FALSE.
Note  When *style-element* is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip  Use the GRIDATTRS= option to control the appearance of the major grid lines.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example  Here is an example that specifies light blue, dotted lines for the minor grid.

```
minorgridattrs=(color=lightblue pattern=dot);
```

**MINORTICKCOUNT=** *positive-integer*

specifies the number of minor ticks that are displayed on the axis.

Interactions  The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

Tip  To display $n$ intervals between major ticks, use MINORTICKCOUNT=$n-1$.

**MINORTICKS=** *TRUE | FALSE*

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

![Minor ticks example](image)

Default  FALSE

Interaction  If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Tip  Use the MINORGRID= option to display grid lines at the minor tick values.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**OFFSETMAX=** *AUTO | AUTOCOMPRESS | number*

reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.

**AUTO**

reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.
AUTOCOMPRESS

applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

number

specifies the offset as a decimal proportion of the full axis length.

Default  AUTO

Range  0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

See  “Adjusting Axis Offsets” on page 985

OFFSETMIN=| AUTO | AUTOCOMPRESS | number

reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.

AUTO

reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS

applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

number

specifies the offset as a decimal proportion of the full axis length.

Default  AUTO

Range  0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

See  “Adjusting Axis Offsets” on page 985.

REVERSE=TRUE | FALSE

specifies whether tick values should appear in the reverse order.

Default  FALSE

See  “boolean ” on page 1479 for other Boolean values that you can use.

SHORTLABEL="string"

specifies an alternate axis label to display when the default label or the label specified by the LABEL= option is too long to fit the available space.

Interaction  This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

Note  If the specified label is itself too long for the axis, it is truncated in the display.

THRESHOLDMAX=number

specifies a bias for including one more tick mark at the maximum end of the axis.

Default  0.30

Range  0–1
Restriction

This option applies to the LAYOUT OVERLAYEQUATED XAXISOPTS= and YAXISOPTS= options only.

Interaction

This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

Tips

If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0.

See

“Adjusting Axis Thresholds” on page 984

**THRESHOLDMIN=**number

specifies a bias for including one more tick mark at the minimum end of the axis.

Default 0.30

Range 0–1

Restriction

This option applies to the LAYOUT OVERLAYEQUATED XAXISOPTS= and YAXISOPTS= options only.

Interaction

This option is ignored if the TICKVALUELIST= or TICKVALUESEQUENCE= option is used.

Tips

If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying THRESHOLDMIN=0 and THRESHOLDMAX=0 prevents the tick marks from extending beyond the data range.

Specifying THRESHOLDMIN=1 and THRESHOLDMAX=1 ensures that the data range is bounded by tick marks.

For the minimum axis length, set the THRESHOLDMIN= and THRESHOLDMAX= options to 0.

See

“Adjusting Axis Thresholds” on page 984

**TICKVALUEATTRS=**style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the axis tick values.

Default The GraphValueText style element.

Interaction

This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick values.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
“Text Options” on page 1525 for available text-options.

**TICKVALUEFITPOLICY=** *policy*

specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

**EXTRACT**

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

**Requirement**
The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*.

**EXTRACTALWAYS**

same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

**Requirement**
The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*.

**NONE**

makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

**SPLIT**

splits the tick value at a split character, which is specified by the **TICKVALUESPLITCHAR=** option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

**See** **TICKVALUESPLITCHAR=**

**SPLITALWAYS**

always splits the axis tick value at every occurrence of a split character that is specified by the **TICKVALUESPLITCHAR=** option.

**See** **TICKVALUESPLITCHAR=**

**SPLITALWAYSTHIN**

same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

**SPLITTHIN**

same as SPLIT, except that thinning is performed when long words do not fit the available space.
THIN
   eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

EXTRACT
   display consecutive integers along the axis instead of the actual tick values to
   represent those tick values.

   In most cases, this policy is implemented if the system estimates that a
   collision might occur. If no collision occurs, then the actual tick values are
   displayed on the axis in the normal manner.

   Requirement  The EXTRACT policy must be used with an AXISLEGEND
                 statement. For more information, see "Extracting Discrete Axis
                 Tick Values into a Legend" in SAS Graph Template Language:
                 User’s Guide.

EXTRACTALWAYS
   same as EXTRACT, except that the extraction is implemented regardless of
   whether collision occurs.

   Requirement  The EXTRACTALWAYS policy must be used with an
                 AXISLEGEND statement. For more information, see "Extracting
                 Discrete Axis Tick Values into a Legend" in SAS Graph

NONE
   does not attempt to fit tick values that collide.

ROTATE
   rotates the tick values if a collision occurs.

ROTATEALWAYS
   rotates the tick values regardless of whether a collision occurs.

ROTATEALWAYSDROP
   attempts the ROTATEALWAYS policy, and then drops the tick values if
   collisions still occur.

ROTATETHIN
   attempts the ROTATE policy, and then the THIN policy.

SPLIT
   splits the tick value at a split character, which is specified by the
   TICKVALUESPLITCHAR= option, only when necessary in order to make the
   value fit the available space.

   A split does not occur at a split character if a split is not needed at that
   location. If the value does not contain any of the specified split characters,
   then the value is not split. Values that are not split or that do not fit the
   available space even after splitting might overlap the adjoining space.

   See  TICKVALUESPLITCHAR=

SPLITALWAYS
   always splits the axis tick value at every occurrence of a split character that is
   specified by the TICKVALUESPLITCHAR= option.

   See  TICKVALUESPLITCHAR=
SPLITROTATE
attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

Note: This feature applies to SAS 9.4M5 and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.
Thinning is performed when the tick values do not fit the available space.

Note: This feature applies to SAS 9.4M5 and to later releases.

STAGGER
alternates the tick values between two rows.

STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.

TRUNCATETHIN
attempts the TRUNCATE policy, and then the THIN policy.

Default THIN

Note A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=(format-options) | DATA | format
specifies how to format the values for major tick marks.

(format-options)
specifies one or more formatting options for major tick values.

Together, these options provide parameters for determining an optimal format (w.d, Ew., BESTw.) for displaying major tick values.

MAXWIDTH=integer
specifies the maximum width for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default 8
**MAXDECIMALS=integer**
specifies the maximum number of decimals for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default 6

**Note**
The MAXWIDTH= option value should be greater than the MAXDECIMALS= option value.

**PREFERREDDECIMALS=integer**
specifies the number of decimal places that you want to display for most values.

The actual number might vary based on other constraints.

Default 2

**EXTRACTSCALE=TRUE | FALSE**
specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.

The scale can be a named scale or a scientific-notation scale. The EXTRACTSCALETYPE= option specifies the scale type. The scale that is used is appended to the axis label, as shown in the following example.

**Total Sales (millions)**

For long axis labels, if the scale does not fit the available space, then the label is truncated, and the scale is appended to the truncated label. Ellipses indicate that the label was truncated, as shown in the following example.

**Total Sales for the Fourth Quarter Of ... (millions)**

In extreme cases in which the scale does not fit even with truncation, the entire axis is dropped.

Default FALSE

**Restriction**
The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale for all locales.

**Interactions**
The scale type is determined by the EXTRACTSCALETYPE= option.

If the axis label is not displayed, then the EXTRACTSCALE=TRUE option is ignored.

**Note**
When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

**See**
"boolean " on page 1479 for other Boolean values that you can use.

**EXTRACTSCALETYPE=DEFAULT | SCIENTIFIC**
specifies whether to extract a named scale or a scientific-notation scale.
DEFAULT
extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as $10^n$) for values over 999 trillion. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of $1/10$ ($10^{-n}$) for values less than 1 trillionth.

SCIENTIFIC
extracts a scientific-notation scale.

A scientific-notation scale is a multiple of 10 expressed as $10^n$ for values greater than 1, or a multiple of $1/10$ expressed as $10^{-n}$ for values less than 1.

Default

Restriction
The scale is derived from the English locale for all locales.

DATA
uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

format
specifies a format to apply to the major tick values.

Restriction
GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

Note
If you specify a format that significantly reduces precision, because of tick-value rounding, the plot data elements might not align properly with the axis tick values. In that case, specify a tick-value format with a higher precision.

Default

VIEWMAX=number
specifies the maximum data value to include in the display on the X or Y axes when the axis lengths and major tick values are equal. The value might be adjusted by the threshold calculation.

Default
The maximum value in the data for the X and Y axes.

Interactions
This option is ignored when EQUATETYPE=SQUARE.

This option does not determine the maximum axis tick value that is displayed. The THRESHOLDMAX= value is used to determine the maximum tick value.

Notes
Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.
The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

**Tip**
To display the VIEWMAX= value as the maximum tick value, use the 
TICKVALUELIST= option.

**See**
“Adjusting the Axis View” on page 983

**VIEWMIN=number**
specifies the minimum data value to include in the display on the X or Y axes when the axis lengths and major tick values are equal. The value might be adjusted by the threshold calculation.

**Default**
The minimum value in the data for the X and Y axes.

**Interactions**
This option is ignored when EQUATETYPE=SQUARE.

This option does not determine the minimum axis tick value that is displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

**Notes**
Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

**Tip**
To display the VIEWMIN= value as the minimum tick value, use the 
TICKVALUELIST= option.

**See**
“Adjusting the Axis View” on page 983

**Details**
The LAYOUT OVERLAYEQUATED statement is used for equated layouts where the X and Y axes always have equal increments between tick values. Because the axes within the equated layout are so closely correlated, some axis adjustments cannot be applied to one axis without applying them to the other.

For example, the INTEGER= option sets evenly spaced integer values for the axis tick marks. That setting must be applied to both axes if the correlation between them is to be maintained. The LAYOUT OVERLAYEQUATED statement provides the COMMONAXISOPTS= option for specifying the INTEGER= option and other options whose settings apply in common to both axes.

Despite the close correlation between the axes, some axis adjustments can be made to one axis without affecting the other. For example, displaying grid lines on one axis has no impact on the other. The XAXISOPTS= and YAXISOPTS= options are available for applying settings separately to the X and Y axes.

The following example template sets evenly spaced integer values for the axis tick marks of both axes. It also specifies the display of grid lines, tick marks, and tick values for the Y axis:
Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL

Axis options for the plots within DATALATTICE and DATAPANEL layouts

See:
“LAYOUT DATALATTICE Statement” on page 47
“LAYOUT DATAPANEL Statement” on page 74

Syntax

The X and X2 axis options for the plots within DATALATTICE and DATAPANEL layouts are specified with the following options:

\[
\text{COLUMNAXISOPTS} = (axis-opt) \\
\text{COLUMN2AXISOPTS} = (axis-opt)
\]

The Y and Y2 axis options for the plots within DATALATTICE and DATAPANEL layouts are specified with the following options:

\[
\text{ROWAXISOPTS} = (axis-opt) \\
\text{ROW2AXISOPTS} = (axis-opt)
\]

General Options for All Axes in the Layout

The options that are documented in this section can be used with any of the axis types that are supported within a DATALATTICE or DATAPANEL layout. Subsequent sections in the chapter document the axis options that are available only for specific axis types: discrete, linear, log, or time axes. The following table provides a summary of the general options.

<table>
<thead>
<tr>
<th>Statement Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTDISPLAY</td>
<td>Controls which axis features are displayed on second, fourth, and other even row or column occurrences of the primary axis.</td>
</tr>
<tr>
<td>ALTDISPLAYSECONDARY</td>
<td>Controls which features are displayed on second, fourth, and other even row or column occurrences of the secondary axis.</td>
</tr>
<tr>
<td>DISCRETEOPTS</td>
<td>Specifies options for a discrete axis.</td>
</tr>
<tr>
<td>Statement Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Controls which axis features are displayed on first, third, and other odd row or column occurrences of the primary axis.</td>
</tr>
<tr>
<td>DISPLAYSECONDARY</td>
<td>Controls which axis features are displayed on first, third, and other odd row or column occurrences of the secondary axis.</td>
</tr>
<tr>
<td>GRIDATTRS</td>
<td>Specifies the attributes of the grid lines.</td>
</tr>
<tr>
<td>GRIDDISPLAY</td>
<td>Specifies whether axis grid lines are displayed.</td>
</tr>
<tr>
<td>LABEL</td>
<td>Specifies the axis label.</td>
</tr>
<tr>
<td>LABELATTRS</td>
<td>Specifies the color and font attributes of the axis label.</td>
</tr>
<tr>
<td>LABELFITPOLICY</td>
<td>Specifies a policy for fitting axis labels in the available space.</td>
</tr>
<tr>
<td>LABELPOSITION</td>
<td>Specifies the position of the axis label.</td>
</tr>
<tr>
<td>LABELSPLITCHAR</td>
<td>Specifies one or more characters on which the axis labels can be split, if needed.</td>
</tr>
<tr>
<td>LABELSPLITCHARDROP</td>
<td>Specifies whether the split characters should be included in the axis labels that are displayed.</td>
</tr>
<tr>
<td>LABELSPLITJUSTIFY</td>
<td>Specifies the justification of the strings that are inside the axis label blocks.</td>
</tr>
<tr>
<td>LINEAROPTS</td>
<td>Specifies options for a standard numeric interval axis.</td>
</tr>
<tr>
<td>LOGOPTS</td>
<td>Specifies options for a log axis.</td>
</tr>
<tr>
<td>NAME</td>
<td>Assigns a name to an axis for reference in other statements.</td>
</tr>
<tr>
<td>OFFSETMAX</td>
<td>Reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>OFFSETMIN</td>
<td>Reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.</td>
</tr>
<tr>
<td>REVERSE</td>
<td>Specifies whether the tick values should appear in the reverse order.</td>
</tr>
<tr>
<td>SHORTLABEL</td>
<td>Specifies an alternate axis label.</td>
</tr>
</tbody>
</table>
### Statement Option Description

**TICKVALUEATTRS**
Specifies the color and font attributes of the axis tick value labels.

**TICKVALUEHALIGN**
Specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.

**TICKVALUEVALIGN**
Specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.

**TIMEOPTS**
Specifies options for a TIME axis.

**TYPE**
Specifies the type of axis to use.

---

**ALTDISPLAY=**
*STANDARD | ALL | NONE | *(display-options)*
controls which axis features are displayed on second, fourth, and other even row or column occurrences of the primary axis.

**STANDARD**
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

**ALL**
currently the same as STANDARD

**NONE**
specifies that no axis features are displayed

*(display-options)*
a space-separated list of one or more of the following options enclosed in parentheses:

**LABEL**
displays the axis label.
A common label is displayed at the center of the grid, and the label applies to all the primary axes in the row or column. This label is overridden by a label specified on the DISPLAY= option.

**LINE**
displays the axis line.

**TICKS**
displays the tick marks.

**TICKVALUES**
displays the values that are represented by the major tick marks.

**Default**
The settings on the DISPLAY= option.

**Tips**
The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

This option can be used to obtain the alternating axis information as seen in a ScatterPlotMatrix.

**See**
“Details” on page 1206 for more information about the primary and secondary axes.
ALTDISPLAYSECONDARY=NONE | ALL | STANDARD | (display-options)
controls which features are displayed on second, fourth, and other even row or column occurrences of the secondary axis. A secondary axis is not an independent axis. Rather, it mirrors the primary axis (though it can use different display features). Thus, for this option to take effect, all plot statements in the LAYOUT PROTOTYPE must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left).

NONE
specifies that no axis features are displayed

STANDARD
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL
currently the same as STANDARD

(display-options)
a space-separated list of one or more of the following options enclosed in parentheses:

LABEL
displays the axis label.
A common label is displayed at the center of the grid, and the label applies to all the secondary axes in the row or column. This label is overridden by a label specified on the DISPLAY= option.

LINE
displays the axis line.

TICKS
displays the tick marks.

TICKVALUES
displays the values that are represented by the major tick marks.

Default
The settings on the DISPLAYSECONDARY= option.

Restriction
If some plot statements set XAXIS=X and others set XAXIS=X2, then both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y axes.

Tips
The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

This option can be used to obtain the alternating axis information as seen in a ScatterPlotMatrix.

See
“Details” on page 1206 for more information about the primary and secondary axes.

GRIDDISPLAY= and GRIDATTRS= for setting axis grid lines.
DISCRETOPTS=(discrete-axis-options)
specifies one or more options for a discrete axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction  This option is ignored if the axis type is not DISCRETE.

See  “Options for Discrete Axes Only” on page 1165 for the options that you can use for discrete-axis-options.

DISPLAY=STANDARD | ALL | NONE | (display-options)
controls which axis features are displayed on first, third, and other odd row or column occurrences of the primary axis.

STANDARD  specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed

ALL  currently the same as STANDARD

NONE  specifies that no axis features are displayed

(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

LABEL  displays the axis label

LINE  displays the axis line

TICKS  displays the tick marks

TICKVALUES  displays the values that are represented by the major tick marks

Default  STANDARD

Note  When LABEL is specified, a common label is displayed at the center of the grid, and the label applies to all the primary axes in the row or column. This label overrides a label specified on the ALTDISPLAY= option.

Tips  The default line attributes for the axis line and axis tick marks are defined in the GraphAxisLine style element.

Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

When LINE is excluded from the DISPLAY= option, the layout wall outline or the default baseline of a bar chart, needle plot, or waterfall chart can appear to be an axis line. To suppress the wall outline, use the WALLDISPLAY= option in the layout statement. To suppress the plot baseline, use the BASELINEATTRS= option in the plot statement.

See  “Details” on page 1206 for more information about the primary and secondary axes.
DISPLAYSECONDARY=NONE | ALL | STANDARD | (display-options)
controls which axis features are displayed on first, third, and other odd row or column occurrences of the secondary axis. A secondary axis is not an independent axis. Rather, it mirrors the primary axis (though it can use different display features). Thus, for this option to take effect, all plot statements in the LAYOUT PROTOTYPE must map data to the same primary axis. For example, a secondary X2 axis can be displayed on top in the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the right in the layout, provided all plot statements set YAXIS=Y to map data to the primary Y axis (left).

NONE
specifies that no axis features are displayed

STANDARD
specifies that the LABEL, LINE, TICKS, and TICKVALUES are displayed on the secondary axis

ALL
currently the same as STANDARD

(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or more of the following options:

LABEL
displays the axis label

LINE
displays the axis line

TICKS
displays the tick marks

TICKVALUES
displays the values that are represented by the major tick marks

Default
 NONE

Restriction
If some plot statements set XAXIS=X and others set XAXIS=X2, both the X and X2 axis are primary and a secondary X axis cannot be displayed. In that case, this option is ignored. The same applies for the Y axes.

Note
When LABEL is specified, a common label is displayed at the center of the grid, and the label applies to all the primary axes in the row or column. This label overrides a label specified on the ALTDISPLAY= option.

Tip
Use the GRIDDISPLAY= and GRIDATTRS= options to set the axis grid lines.

See
for more information about the primary and secondary axes. “Details” on page 1206.

GRIDATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the grid lines.

Default
The GraphGridLines style element.
Interaction

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tip

On a log axis, this option affects the appearance of the major grid lines only. It does not affect the appearance of the minor grid lines. To control the appearance of the minor grid lines on a log axis, use the MINORGRIDATTRS= option.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

GRIDDISPLAY=AUTO_OFF | AUTO_ON | ON | OFF

specifies whether axis grid lines are displayed. This option enables the template to absolutely control the display of grid lines or to allow interaction with the current style to decide whether grid lines are displayed.

AUTO_OFF

specifies that grid lines are not displayed unless the GraphGridLines element in the current style contains DisplayOpts="ON."

AUTO_ON

specifies that grid lines are displayed unless the GraphGridLines element in the current style contains DisplayOpts="OFF."

ON

specifies that grid lines are always displayed.

The current style has no override.

OFF

specifies that grid lines are never displayed.

The current style has no override.

The following table shows the end results for various combinations of the GRIDDISPLAY= option and the DisplayOpts= attribute of the GraphGridLines style element. Most supplied templates use the default setting AUTO_OFF to indicate a preference for not displaying grid lines, but allowing the style to override.

<table>
<thead>
<tr>
<th>GRIDDISPLAY= option</th>
<th>DisplayOpts= style attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_OFF</td>
<td>AUTO</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_OFF</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>AUTO</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>ON</td>
<td>yes</td>
</tr>
<tr>
<td>AUTO_ON</td>
<td>OFF</td>
<td>no</td>
</tr>
<tr>
<td>ON</td>
<td>any value</td>
<td>yes</td>
</tr>
</tbody>
</table>
### GRIDDISPLAY= Option

<table>
<thead>
<tr>
<th>DisplayOptions= Style Attribute</th>
<th>Grid Lines Shown?</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>any value</td>
</tr>
</tbody>
</table>

**Default**  AUTO_OFF

**Note** Supplied styles use DisplayOptions="AUTO," which means that the style has no preference about grid lines and the graphics template setting for grid lines is always used.

**LABEL=** "string" | ("string" ... "string")

specifies the axis label. The string can be either a string literal or a dynamic. The list form implies that all included string literals or dynamic variables will be concatenated.

**Default** The default label is derived from the primary plot in the layout. For more information, see “When Plots Share Data and a Common Axis” on page 979.

**Interaction** This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

**Note** If the axis label is too long to fit along the axis, then it is truncated by default.

**Tips** Use the SHORTLABEL= option to specify an alternate axis label to be used whenever truncation would normally occur.

Use the ODS escape sequence to specify Unicode characters in an axis label. Here is an example that specifies degrees Fahrenheit in an axis label for temperature:

```
label="Temperature \{\text*\{Unicode \text*\{00B0\x{7F}\}'F}"
```

**See** “Overriding the Default Axis Label” in SAS Graph Template Language: User’s Guide

**LABELATTRS=** style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the axis label.

**Default** The GraphLabelText style element.

**Interaction** This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display the axis label.

**See** “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**LABELFITPOLICY=** AUTO | SPLITALWAYS | STACKED

specifies a policy for fitting axis labels in the available space.

**AUTO** uses the short label, when specified, instead of the original label.
If the short label does not fit, then it is clipped. When no short label is specified, the original label is clipped.

**SPLIT**

splits the axis label at a split character only when necessary in order to make the label fit the available space.

The split character is specified by the LABELSPLITCHAR= option. The short label is not used. A split does not occur at a split character if a split is not needed at that location. If the label does not contain any of the specified split characters, then it is not split. A label that cannot be split or that does not fit the available space even after splitting might overlap the adjoining space.

**SPLITALWAYS**

always split the axis label at every occurrence of a split character, which is specified by the LABELSPLITCHAR= option.

If the label cannot be split, then it is clipped.

**STACKED**

displays the Y or Y2 axis label vertically as stacked letters.

---

**Note:** This feature applies to **SAS 9.4M5** and to later releases.

---

Valid in **ROWAXISOPTS=** and **ROW2AXISOPTS=** only

---

**Default** AUTO

**Interaction** This option has effect only when LABELPOSITION= is CENTER or DATACENTER.

**Note** When LABELPOSITION=CENTER, the available area is the full axis, including the axis offsets. When LABELPOSITION=DATACENTER, the available area is the tick display area, excluding the axis offsets.

**LABELPOSITION**

specifies the position of the axis label.

**CENTER**

centers the axis label in the axis area.

For the Y and Y2 axes, the label is oriented vertically and is centered in the axis area (including the offsets). The label is positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis. For the X and X2 axes, the label is centered in the axis area (including the offsets). It is positioned below the tick values for the X axis or above the axis values for the X2 axis.

**DATACENTER**

repeats the axis label for each row or column and centers each label in the axis tick display area of its row or column.

For the Y and Y2 axes, each label is oriented vertically and is centered in the axis tick display area (excluding the offsets) of its row. The labels are positioned to the left of the tick values for the Y axis or to the right of the axis values for the Y2 axis. For the X and X2 axes, each label is centered in the axis tick display area (excluding the offsets) of its column. The labels are positioned below the tick values for the X axis or above axis values for the X2 axis.
TOP

orients the label horizontally at the top or bottom of the axis area.

The label is right-justified in the axis area for the Y axis and left-justified for the Y2 axis. If there is not sufficient room in the axis area to display the label, then the label grows to the right for the Y axis and to the left for the Y2 axis.

Restriction These options are valid for the Y and Y2 axes only.

Note When TOP or BOTTOM is used, the label might collide with other graphical features. In that case, use CENTER or DATACENTER instead.

LEFT

RIGHT

positions the label to the left or right of the axis area.

The label is centered vertically in the axis area.

Restriction These options are valid for the X and X2 axes only.

Note When LEFT or RIGHT is used, the label might collide with other graphical features.

The following figure shows the CENTER and DATACENTER positions for a blue Y-axis label Type and a red X-axis label MPG.

The next figure shows the TOP and LEFT positions, and the BOTTOM and RIGHT positions for the same axis labels.

The following figure shows the CENTER and DATACENTER positions for a blue Y-axis label Qtr and a red X-axis label Close.

The next figure shows the TOP and LEFT positions, and the BOTTOM and RIGHT positions for the same axis labels.
Restriction: This option does not support collision avoidance. In some cases, axis label collisions can occur in the axis area.

Interaction: When LEFT, RIGHT, TOP, or BOTTOM is in effect, the SHORTLABEL= option is ignored.

See SHORTLABEL= on page 1162 for information about how short labels are used.

**LABELSPLITCHAR=**"character-list"

This option specifies one or more characters on which the axis labels can be split, if needed. When multiple split characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the axis label. In that case, all of the specified split characters together are treated as a single split character.

When LABELFITPOLICY=SPLIT, if the axis label does not fit the available space, then it is split on a specified split character only if a split is needed at that point to make the label fit. In this case, a split might not occur on every split character. When LABELFITPOLICY=SPLITALWAYS, the axis label is split unconditionally on every occurrence of a split character. If the axis label does not contain any of the specified split characters, the label is not split.

"character-list"

One or more characters with no space between each character and enclosed in quotation marks.

Default: A blank space

Requirements: The list of characters must be enclosed in quotation marks.

Multiple characters must be specified with no space between them. For example, to specify the split characters a, b, and c, use the following option:

labelsplitchar="abc"
### Interactions

This option has effect only when `LABELPOSITION=` is CENTER or DATACENTER.

The `LABELSPLITCHARDROP=` option specifies whether the split characters are included in the displayed data label or are dropped.

### Notes

When multiple characters are specified, the order of the characters in the list is not significant.

The split characters are case sensitive.

### Tip

Use the `LABELSPLITJUSTIFY=` option to specify the justification of the strings in the axis label block.

---

**LABELSPLITCHARDROP=TRUE | FALSE**

specifies whether the split characters are included in the displayed axis labels.

<table>
<thead>
<tr>
<th>TRUE</th>
<th>drops the split characters from the axis label display.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALSE</td>
<td>includes the split characters in the axis label display.</td>
</tr>
</tbody>
</table>

When the label is split while `LABELSPLITCHARDROP=FALSE` is in effect, each split character remains as the last character in the current line. The characters that follow the split character, up to and including the next split character, are then wrapped to the next line.

**Default**

TRUE. The split characters are dropped from the axis label.

### Interactions

This option has effect only when `LABELPOSITION=` is CENTER or DATACENTER.

The `LABELSPLITCHAR=` option specifies the split characters.

See “boolean” on page 1479 for other Boolean values that you can use.

---

**LABELSPLITJUSTIFY=justification**

specifies the justification of the strings that are inside the axis label blocks.

<table>
<thead>
<tr>
<th>justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>CENTER</td>
</tr>
<tr>
<td>LEFT</td>
</tr>
<tr>
<td>RIGHT</td>
</tr>
</tbody>
</table>

specifies the justification for the X or X2 axis label.

<table>
<thead>
<tr>
<th>CENTER</th>
<th>TOP</th>
<th>BOTTOM</th>
</tr>
</thead>
</table>

specifies the justification for the Y or Y2 axis label.

**Default**

CENTER

### Interaction

This option has effect only when `LABELPOSITION=` is CENTER or DATACENTER.
LINEAROPTS=(linear-axis-options)
specifies one or more options for a numeric interval axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction  This option is ignored if the axis type is not LINEAR.

See  “Options for Linear Axes Only” on page 1173 for the options that you can use for linear-axis-options.

LOGOPTS=(log-axis-options)
specifies one or more options for a log axis. Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction  This option is ignored if the axis type is not LOG.

See  “Options for Log Axes Only” on page 1186 for the options that you can use for log-axis-options.

NAME="string"
assigns a name to an axis for reference in other statements. Currently, it is used only in an AXISLEGEND statement.

Restriction  This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Interactions  This option is ignored unless the axis is discrete. The axis can be discrete by default, or explicitly set to discrete with a TYPE=DISCRETE setting.

For this option to take effect, an axis legend must be enabled. To enable an axis legend, the DISCRETEOPTS= option must set the TICKVALUEFITPOLICY to either EXTRACT or EXTRACTALWAYS. In addition, an AXISLEGEND statement must be specified to generate the axis legend.

OFFSETMAX=AUTO | AUTOCOMPRESS | number
reserves an area at the maximum end of the axis. No tick marks are displayed in the reserved area.

AUTO  reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS  applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

number  specifies the offset as a decimal proportion of the full axis length.

For a continuous axis, the offset follows the highest data value or highest tick value, whichever is greater.

Default  AUTOCOMPRESS

Range  0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.
OFFSETMIN=AUTO | AUTOCOMPRESS | number
reserves an area at the minimum end of the axis. No tick marks are displayed in the reserved area.

AUTO
reserves just enough area to fully display markers and other graphical features near the maximum end of an axis.

AUTOCOMPRESS
applies an automatic offset that prevents axis labels and tick values from extending beyond the axis length.

number
specifies the offset as a decimal proportion of the full axis length.

For a continuous axis, the offset precedes the lowest data value or lowest tick value, whichever is less.

Default AUTOCOMPRESS

Range 0–1. The sum of OFFSETMAX= and OFFSETMIN= should not be more than 1.

See “Adjusting Axis Offsets” on page 985.

REVERSE=TRUE | FALSE
specifies whether tick values should appear in the reverse order.

Default FALSE

See “boolean ” on page 1479 for other Boolean values that you can use.

SHORTLABEL="string"
specifies an alternate axis label to display when the default label or the label specified by the LABEL= option is too long to fit the available space. When LABELPOSITION=CENTER (default), the available space for an axis label is the full axis, including the axis offsets. When LABELPOSITION=DATA CENTER, the available space for an axis label is the axis tick display area, which excludes the axis offsets. If the label length exceeds the available space, then the label is anchored at the left or bottom offset. It extends beyond the opposing offset until it reaches the end of the axis where it is truncated. An ellipsis designates the truncation.

Interactions This option is ignored if the DISPLAY= option or the DISPLAYSECONDARY= option does not display the axis label.

This option has effect only when the LABELPOSITION= option is set to CENTER or DATACENTER.

Note If the specified label is itself too long for the grid length or the grid width, then it is truncated in the display.

TICKVALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the axis tick values.

Default The GraphValueText style element.
Interaction  This option is ignored if the DISPLAY= or DISPLAYSECONDARY= option does not display tick values.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

\textbf{TICKVALUEHALIGN=LEFT | CENTER | RIGHT}  
specifies the horizontal alignment for all of the tick values that are displayed on the Y and Y2 axes.

\begin{center}
\begin{tabular}{c|c|c}
LEFT & CENTER & RIGHT \\
\hline 
Notebooks & Notebooks & Notebooks \\
Ink & Ink & Ink \\
Printer paper & Printer paper & Printer paper \\
Staples & Staples & Staples \\
Pens & Pens & Pens \\
\end{tabular}
\end{center}

Defaults  RIGHT for a Y axis

LEFT for a Y2 axis

Restriction  This option is valid for the Y and Y2 axes only.

\textbf{TICKVALUEVALIGN=TOP | CENTER | BOTTOM}  
specifies the vertical alignment for all of the tick values that are displayed on the X and X2 axes.

\begin{center}
\begin{tabular}{c|c|c|c|c|c}
TOP & CENTER & BOTTOM & TOP & CENTER & BOTTOM \\
\hline 
Stapler & Paper & Calendar & Toner & Large & Desk \\
Clips & Cartridges & Blotters & Clips & Cartridges & Blotters \\
\end{tabular}
\end{center}

Defaults  TOP for an X axis

BOTTOM for an X2 axis

Restriction  This option is valid for the X and X2 axes only.

\textbf{TIMEOPTS=(time-axis-options)}  
specifies one or more options for a time axis.
Requirements

Columns associated with a time axis must be in SAS time, SAS date, or SAS datetime units and have an associated SAS time, date, or datetime format.

Options must be enclosed in parentheses. Each option is specified as a name = value pair and each pair is space separated.

Interaction

This option is ignored if the axis type is not TIME.

See

“Options for Time Axes Only” on page 1195 for the options that you can use for time-axis-options.

**TYPE=**AUTO | DISCRETE | LINEAR | TIME | LOG

specifies the type of axis to use.

**AUTO**

requests that the axis type be automatically determined, based on the overlay contents.

**DISCRETE**

uses a DISCRETE axis if possible.

The data for discrete axes can be character or numeric. You can add a DISCRETEOPTS= option list to customize this axis type.

**LINEAR**

uses a LINEAR axis if possible.

You can add a LINEAROPTS= option list to customize this axis type.

**TIME**

uses a TIME axis if possible.

Data for this axis must be SAS time, SAS date, or SAS datetime values. You can add a TIMEOPTS= option list to customize this axis type.

**LOG**

uses a LOG axis if possible.

You can add a LOGOPTS= option list to customize this axis type.

Interaction

If a log axis is requested and the axis data contains 0 or negative values, the axis reverts to a linear axis. This outcome can occur for the response axis of a bar chart, line chart, needle plot, or waterfall chart when a baseline intercept of 0 or less is specified. It can also occur for the response axis of a waterfall chart when an initial bar value of 0 or less is specified. To get a log response axis in those cases, set the baseline intercept or initial bar value to a positive value.

Default

AUTO

Interactions

If this option is set to anything other than AUTO, then plots within the layout are dropped from the display if their data types or data ranges do not match the axis type requirements. For more information, see “Plot Axis Types Must Agree on Common Axes” on page 981.

After the axis type is determined (whether you set a specific type or AUTO is in effect), you can use only options that are supported by
that axis type. For example, if TYPE=TIME, then only the general OVERLAY axis options and those available on TIMEOPTS= are supported.

Options for Discrete Axes Only

The options that are documented in this section can be used with the DISCRETEOPTS= axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Discrete Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TICKDISPLAYLIST</td>
<td>Specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>TICKTYPE</td>
<td>Specifies the position of the axis tick mark.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the list of tick values that are displayed on the axis.</td>
</tr>
<tr>
<td>TICKVALUEROTATION</td>
<td>Specifies how the tick values are rotated on the X and X2 axes.</td>
</tr>
<tr>
<td>TICKVALUESPLITCHAR</td>
<td>Specifies a list of characters on which the tick values can be split, if needed.</td>
</tr>
<tr>
<td>TICKVALUESPLITCHARDROP</td>
<td>Specifies whether the split characters are included in the displayed tick values.</td>
</tr>
<tr>
<td>TICKVALUESPLITJUSTIFY</td>
<td>Specifies justification of the strings that are inside the tick value block.</td>
</tr>
</tbody>
</table>

**TICKDISPLAYLIST=(string-list)**

specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the TICKVALUELIST= option. The strings map one-to-one positionally with the values that are listed in the TICKVALUELIST= option.

Default: Determined by the system or by the TICKVALUELIST= option.

Requirements: The list of values must be enclosed in parentheses. Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.
Tip This option should be used with the **TICKVALUENLIST**= option. The number of items in the list for this option should equal the number of items in the list for the **TICKVALUENLIST**= option.

Example The following example specifies the axis tick values 10, 20, 30, and 40, and the tick display values A, B, C, and D:

```sas
tickvaluelist = (*10* *20* *30* *40*);
tickdisplaylist = (*A* *B* *C* *D*);
```

**TICKTYPE**=MIDPOINT | INBETWEEN

specifies the position of the axis tick marks.

**MIDPOINT**

places the tick marks at the midpoint value location.

**INBETWEEN**

places the tick marks half way between adjacent midpoint locations.

Default MIDPOINT

Restriction This option applies to discrete axes only.

Note Starting with SAS 9.4M2, when **TICKTYPE**=INBETWEEN, the outermost tick marks and grid lines at each end of the axis are not drawn.

**TICKVALUEFITPOLICY**=policy

specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

**EXTRACT**

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

| Requirement | The **EXTRACT** policy must be used with an **AXISLEGEND** statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*. |

**EXTRACTALWAYS**

same as **EXTRACT**, except that the extraction is implemented regardless of whether collision occurs.

| Requirement | The **EXTRACTALWAYS** policy must be used with an **AXISLEGEND** statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*. |

**NONE**

makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.
SPLIT
splits the tick value at a split character, which is specified by the
TICKVALUESPLITCHAR= option, only when necessary in order to make the
value fit the available space.

A split does not occur at a split character if a split is not needed at that
location. If the value does not contain any of the specified split characters,
then the value is not split. Values that are not split or that do not fit the
available space even after splitting might overlap the adjoining space.

See TICKVALUESPLITCHAR=

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is
specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words
do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit
the available space.

THIN
eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

EXTRACT
display consecutive integers along the axis instead of the actual tick values to
represent those tick values.

In most cases, this policy is implemented if the system estimates that a
collision might occur. If no collision occurs, then the actual tick values are
displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND
statement. For more information, see "Extracting Discrete Axis
Tick Values into a Legend" in SAS Graph Template Language:
User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of
whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an
AXISLEGEND statement. For more information, see "Extracting
Discrete Axis Tick Values into a Legend" in SAS Graph

NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.
The `TICKVALUEROTATION=` option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

**ROTATEALWAYS**
rotates the tick values regardless of whether a collision occurs.

The `TICKVALUEROTATION=` option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

**ROTATEALWAYSDROP**
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.

**ROTATETHIN**
attempts the ROTATE policy, and then the THIN policy.

**SPLIT**
splits the tick value at a split character, which is specified by the `TICKVALUESPLITCHAR=` option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See  `TICKVALUESPLITCHAR=`

**SPLITALWAYS**
always splits the axis tick value at every occurrence of a split character that is specified by the `TICKVALUESPLITCHAR=` option.

See  `TICKVALUESPLITCHAR=`

**SPLITROTATE**
attends the SPLIT policy, and then the ROTATE policy.

**STACKEDALWAYS**
always displays the tick values vertically as stacked letters.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**STACKEDALWAYSTHIN**
always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**STAGGER**
alternates the tick values between two rows.

**STAGGERROTATE**
attends the STAGGER policy, and then the ROTATE policy.

**STAGGERTHIN**
attends the STAGGER policy, and then the THIN policy.
STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.

TRUNCATETHIN
attempts the TRUNCATE policy, and then the THIN policy.

Defaults
ROTATE for the X and X2 axes
THIN for the Y and Y2 axes

Note
A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=\text{format}
specifies how to format the values for major tick marks.

Note: This feature applies to SAS 9.4M3 and to later releases.

Restrictions
This option applies only to discrete axes.

Only character formats are supported.

Interaction
This option is ignored when the axis tick values are extracted to an axis legend. See \text{TICKVALUEFITPOLICY=EXTRACT | EXTRACTALWAYS on page 1166.}

Tip
Use this option when you want to duplicate tick values on an axis.

TICKVALUELIST=(\text{string-list})
specifies the list of tick values that are to be displayed on the axis.

\text{string-list}
a space-separated list of values, enclosed in parentheses.

You must enclose each value in the list in quotation marks.

Only the tick values that are included in the string list are displayed on the axis. The values are displayed in the order in which they are listed. The data values that are not in the list are dropped. The list can be a subset of the data values. It can also contain values that are not included in the actual data. A tick value that is not included in the data appears on the axis, but no data is represented at its tick mark.

Requirements
The list of values must be enclosed in parentheses.

Each value must be enclosed in quotation marks and separated from adjacent values by a blank space.
If the string list contains duplicate values, then the first occurrence of the duplicated value in the list is honored and the remaining instances are ignored.

When the values specified in the list are compared with the actual data values, leading blanks are honored and trailing blanks are ignored.

You can use this option to subset the axis values or to display the values in a specific order.

You can use this option to display values on the axis that are not contained in the data.

The following example specifies the axis tick values Sedan, Sports, Wagon, and SUV:

```
tickvaluelist=لاق"Sedan" "Sports" "Wagon" "SUV"
```

The following example specifies the axis tick values 10, 20, 30, and 40:

```
tickvaluelist=لاق"10" "20" "30" "40"
```

**TICKVALUEROTATION=** **DIAGONAL | DIAGONAL2 | VERTICAL**

specifies how the tick values are rotated on the X and X2 axes.

**DIAGONAL**

rotates the tick values to a 45-degree diagonal position.

The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

**DIAGONAL2**

rotates the tick values to a –45-degree diagonal position.

The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

Note: This feature applies to **SAS 9.4M5** and to later releases.

**VERTICAL**

rotates the labels to a 90-degree vertical position.

The labels are always drawn from bottom to top.

**Default**

**DIAGONAL**

**Restriction**

This option is valid for **COLUMNAXISOPTS=** and **COLUMN2AXISOPTS=** only.

**Interaction**

The **TICKVALUEFITPOLICY=** option must be set to **ROTATE** or **ROTATEALWAYS** for this option to have any effect.

**TICKVALUESPLITCHAR="character-list"**

specifies a list of characters on which the tick values can be split, if needed. When multiple characters are specified, each character in the list is treated as a separate split character unless the specified characters appear consecutively in the tick value. In that case, all of the specified split characters together are treated as a single split character.
When TICKVALUESPLITPOLICY=SPLIT, if a tick value collision is detected, then the tick value is split on a split character only if necessary at that point in order to avoid collision. In that case, a split might not occur on every split character. When TICKVALUEFITPOLICY=SPLITALWAYS, the tick value is split unconditionally on every occurrence of a split character. If the tick value does not contain any of the specified split characters, then it is not split.

"character-list"
  one or more characters with no delimiter between each character.

Default
  A blank space

Requirements
  The list of characters must be enclosed in quotation marks.
  Multiple characters must be specified with no delimiters. For example, to specify the characters a, b, and c, use the following option:
  
  \[ tickvaluesplitchar="abc" \]

Interactions
  This option is ignored unless option TICKVALUEFITPOLICY= is set to SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.

Notes
  When multiple characters are specified, the order of characters in the list is not significant.
  The split characters are case sensitive.

Tips
  Use the \texttt{TICKVALUESPLITJUSTIFY=} option to specify the justification of the strings in the tick value block.
  For the X and X2 axis tick values, use the \texttt{TICKVALUEVALIGN=} option to specify the vertical alignment of the tick values.
  For the Y and Y2 axis tick values, use the \texttt{TICKVALUEHALIGN=} option to specify the horizontal alignment of the tick values.

Example
  The following example specifies a blank space, a comma, and an underscore as split characters:
  
  \[ tickvaluesplitchar=" ,_" \]

\texttt{TICKVALUESPLITCHARDROP=TRUE | FALSE}

specifies whether the split characters should be included in the displayed tick values. The split characters are specified by the \texttt{TICKVALUESPLITCHAR=} option.

\texttt{TRUE}
  drops the split characters from the tick value display.

  The following figure shows an example in which \texttt{TICKVALUESPLITCHARDROP=TRUE} and three-word, asterisk-delimited tick values are split on the asterisk character by using the SPLITALWAYS policy.
FALSE includes the split characters in the tick value display.

The fit policy determines how the characters are displayed. If the display policy is SPLIT or SPLITTHIN and TICKVALUESPLITCHARDROP=FALSE, then each tick value is split at a split character only where a split is necessary in order to make the value fit the available space. A split might not occur at every split character in the tick value. At each split point, the split character remains as the last character in the current line. The characters that follow the split character, up to and including the split character at the next split point, are then wrapped to the following line. This process repeats until the entire data tick value is displayed. The following figure shows an example in which TICKVALUESPLITCHARDROP=FALSE and three-word, asterisk-delimited tick values are split on the asterisk character by using the SPLIT policy.

Notice that a split occurs on the first asterisk and not at the second. In this case, a split is not needed at the second asterisk.

If the fit policy is SPLITALWAYS or SPLITALWAYSTHIN, and TICKVALUESPLITCHARDROP=FALSE, then each tick value is split at every instance of a split character in the value regardless of whether a split is actually needed. Each split character remains as the last character in the current line. The characters that follow each split character, up to and including the next split character, are then wrapped to the next line. The following figure shows an example in which TICKVALUESPLITCHARDROP=FALSE and three-word, asterisk-delimited tick values are split on the asterisk character by using the SPLITALWAYS policy.

Notice that a split occurs after each asterisk and each asterisk appears at the end of the line. In this case, three lines are displayed.

Default TRUE

Interactions The TICKVALUESPLITCHAR= option specifies the split character or characters.

This option is ignored unless option TICKVALUEFITPOLICY= is set to SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.

See “boolean” on page 1479 for other Boolean values that you can use.
**TICKVALUESPLITJUSTIFY=** CENTER | LEFT | RIGHT
specifies justification of the strings that are inside the tick value block. The justification is relative to an individual tick value's display area and does not affect the display of tick values that are not split.

**Defaults**
- CENTER for an X or X2 axis
- RIGHT for a Y axis
- LEFT for a Y2 axis

**Interaction**
This option is ignored unless option **TICKVALUEFITPOLICY=** is set to SPLIT, SPLITALWAYS, SPLITTHIN, or SPLITALWAYSTHIN.

### Options for Linear Axes Only

The options that are documented in this section can be used with the **LINEAROPTS=** axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Linear Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEGER</td>
<td>Specifies that evenly spaced integer values are used for tick marks.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKCOUNT</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether the minor tick marks are displayed on the axis.</td>
</tr>
<tr>
<td>THRESHOLDMAX</td>
<td>Specifies a bias for including one more tick mark at the maximum end of the axis.</td>
</tr>
<tr>
<td>Linear Axis Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>THRESHOLDMIN</td>
<td>Specifies a bias for including one more tick mark at the minimum end of the axis.</td>
</tr>
<tr>
<td>TICKDISPLAYLIST</td>
<td>Specifies the text that is displayed for the tick values that are defined in the TICKVALUELIST= option.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision. Only the default policy (THIN) is available for a Y or Y2 axis.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values for a linear axis as list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification can extend the axis data range.</td>
</tr>
<tr>
<td>TICKVALUEROTATION</td>
<td>Specifies how the tick values are rotated on the X and X2 axes.</td>
</tr>
<tr>
<td>TICKVALUESEQUENCE</td>
<td>Specifies the tick values for a linear axis by start, end, and increment.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display.</td>
</tr>
</tbody>
</table>

**INTEGER=TRUE | FALSE**

specifies that evenly spaced integer values are used for tick marks.

- **Default**: FALSE

- **Interactions**: This option is overridden by the TICKVALUELIST= or TICKVALUESEQUENCE= option.

  This option overrides the MAXDECIMALS= and PREFERREDDECIMALS= suboptions of the TICKVALUEFORMAT= option.

  INTEGER=TRUE is ignored for the X or X2 axis when a histogram plot is the primary plot and BINAXIS=TRUE is specified in the HISTOGRAMPARM statement.

- **See**: “boolean” on page 1479 for other Boolean values that you can use.

**MINORGRID=TRUE | FALSE**

specifies whether grid lines are displayed at the minor tick marks.
Defaults

FALSE in SAS 9.4M1 and earlier releases.

The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpt is not defined in the active style, then FALSE is the default value.

Interaction

This option is ignored if the GRIDDISPLAY= option does not display the grid lines.

Tips

The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See

“boolean ” on page 1479 for other Boolean values that you can use.

MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults

The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction

This option is ignored when MINORTICKS=FALSE.
Note  When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip  Use the GRIDATTRS= option to control the appearance of the major grid lines.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example  Here is an example that specifies light blue, dotted lines for the minor grid.

minorgridattrs=(color=lightblue pattern=dot);

MINORTICKCOUNT=positive-integer

specifies the number of minor ticks that are displayed on the axis.

Defaults  Four ticks with five intervals in SAS 9.4M1 and earlier releases.

One tick with two intervals starting with SAS 9.4M2.

Interactions  The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

Tip  To display \( n \) intervals between major ticks, use MINORTICKCOUNT=\( n-1 \).

MINORTICKS=TRUE | FALSE

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

```
MINORTICKS=TRUE

TRUE

$0  $50  $100  $150  $200

FALSE

$0  $50  $100  $150  $200
```

Default  FALSE

Interaction  If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform tick values such as 3, 5, 11, and 23, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform tick values such as 0, 10, 20, and 30 in the TICKVALUELIST= option.

Tip  Use the MINORGRID= option to display grid lines at the minor tick values.

See  “boolean” on page 1479 for other Boolean values that you can use.

THRESHOLDMAX=number

specifies a bias for including one more tick mark at the maximum end of the axis.
**Default**
0.30

**Range**
0–1

**Restriction**
This option applies to linear axes only.

**Interaction**
This option is ignored if the `TICKVALUELIST=` or `TICKVALUESEQUENCE=` option is used.

**Tips**
If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying `THRESHOLDMIN=0` and `THRESHOLDMAX=0` prevents the tick marks from extending beyond the data range.

Specifying `THRESHOLDMIN=1` and `THRESHOLDMAX=1` ensures that the data range is bounded by tick marks.

For the minimum axis length, set the `THRESHOLDMIN=` and `THRESHOLDMAX=` options to 0.

**See**
“Adjusting Axis Thresholds” on page 984

**THRESHOLDMIN=** *number*

specifies a bias for including one more tick mark at the minimum end of the axis.

**Default**
0.30

**Range**
0–1

**Restriction**
This option applies to linear axes only.

**Interaction**
This option is ignored if the `TICKVALUELIST=` or `TICKVALUESEQUENCE=` option is used.

**Tips**
If the threshold is set to 0, the potential tick mark is never displayed. If the threshold is set to 1, then the tick mark is always displayed.

Specifying `THRESHOLDMIN=0` and `THRESHOLDMAX=0` prevents the tick marks from extending beyond the data range.

Specifying `THRESHOLDMIN=1` and `THRESHOLDMAX=1` ensures that the data range is bounded by tick marks.

For the minimum axis length, set the `THRESHOLDMIN=` and `THRESHOLDMAX=` options to 0.

**See**
“Adjusting Axis Thresholds” on page 984

**TICKDISPLAYLIST=(** *string-list*)

specifies the text that is displayed for the tick values that are defined in the `TICKVALUELIST=` option. The string list is a space-separated list of string values that are displayed on the axis in place of the values in the `TICKVALUELIST=` option. The strings map one-to-one positionally with the values that are listed in the `TICKVALUELIST=` option.

**Default**
The display of tick values is controlled by the `TICKVALUEFORMAT=` option.
Requirements

The list of values must be enclosed in parentheses.

Each value (character and numeric) must be enclosed in quotation marks and separated from adjacent values by a blank space.

Interaction

When this option is specified, the TICKVALUEFORMAT= option is ignored.

Tip

This option should be used with the TICKVALUELIST= option.
The number of items in the list for this option should equal the number of items in the list for the TICKVALUELIST= option.

TICKVALUEFITPOLICY=\textit{policy}

specifies a policy for avoiding tick value collision on an axis. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. Which policies are valid depends on the axis on which this option is used. For the Y and Y2 axes, the following policies are valid:

\textbf{EXTRACT}

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

\begin{tabular}{|p{0.2\textwidth}|p{0.7\textwidth}|}
\hline
\textbf{Requirement} & The \textit{EXTRACT} policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in \textit{SAS Graph Template Language: User’s Guide}. \\
\hline
\end{tabular}

\textbf{EXTRACTALWAYS}

same as \textit{EXTRACT}, except that the extraction is implemented regardless of whether collision occurs.

\begin{tabular}{|p{0.2\textwidth}|p{0.7\textwidth}|}
\hline
\textbf{Requirement} & The \textit{EXTRACTALWAYS} policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in \textit{SAS Graph Template Language: User’s Guide}. \\
\hline
\end{tabular}

\textbf{NONE}

makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

\textbf{SPLIT}

splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

\begin{tabular}{|p{0.3\textwidth}|p{0.7\textwidth}|}
\hline
\textbf{See} & TICKVALUESPLITCHAR= \\
\hline
\end{tabular}
SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit the available space.

THIN
eliminates alternate tick values.

For the X and X2 axes, the following policies are valid:

EXTRACT
display consecutive integers along the axis instead of the actual tick values to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.
ROTATETHIN
attempts the ROTATE policy, and then the THIN policy.

SPLIT
splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See TICKVALUESPLITCHAR=

SPLITALWAYS
always splits the axis tick value at every occurrence of a split character that is specified by the TICKVALUESPLITCHAR= option.

See TICKVALUESPLITCHAR=

SPLITROTATE
attempts the SPLIT policy, and then the ROTATE policy.

STACKEDALWAYS
always displays the tick values vertically as stacked letters.

Note: This feature applies to SAS 9.4M5 and to later releases.

STACKEDALWAYSTHIN
always displays the tick values vertically as stacked letters.

Thinning is performed when the tick values do not fit the available space.

Note: This feature applies to SAS 9.4M5 and to later releases.

STAGGER
alternates the tick values between two rows.

STAGGERROTATE
attempts the STAGGER policy, and then the ROTATE policy.

STAGGERTHIN
attempts the STAGGER policy, and then the THIN policy.

STAGGERTRUNCATE
attempts the STAGGER policy, and then the TRUNCATE policy.

THIN
eliminates alternate tick values.

TRUNCATE
shortens the tick values when they exceed a certain number of characters.

TRUNCATEROTATE
attempts the TRUNCATE policy, and then the ROTATE policy.

TRUNCATESTAGGER
attempts the TRUNCATE policy, and then the STAGGER policy.
TRUNCATETHIN attempts the TRUNCATE policy, and then the THIN policy.

Default THIN

Note A note is written to the SAS log when tick value thinning occurs.

TICKVALUEFORMAT=(format-options) | DATA | format
specifies how to format the values for major tick marks.

(format-options)
specifies one or more formatting options for major tick values.

Together, these options provide parameters for determining an optimal format (w.d, Ew., BESTw.) for displaying major tick values.

MAXWIDTH=integer
specifies the maximum width for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default 8

MAXDECMALS=integer
specifies the maximum number of decimals for displayed tick values.

Values might be rounded or converted to E notation to fit into this width.

Default 6

Note The MAXWIDTH= option value should be greater than the MAXDECMALS= option value.

PREFERREDDECMALS=integer
specifies the number of decimal places that you want to display for most values.

The actual number might vary based on other constraints.

Default 2

EXTRACTSCALE=TRUE | FALSE
specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.

The scale can be a named scale or a scientific-notation scale. The EXTRACTSCALETYPE= option specifies the scale type. The scale that is used is appended to the axis label, as shown in the following example.

Total Sales (millions)

For long axis labels, if the scale does not fit the available space, then the label is truncated, and the scale is appended to the truncated label. Ellipses indicate that the label was truncated, as shown in the following example.

Total Sales for the Fourth Quarter Of ... (millions)

In extreme cases in which the scale does not fit even with truncation, the entire axis is dropped.

Default FALSE
Restriction | The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale for all locales.

Interactions | The scale type is determined by the EXTRACTSCALETYPE= option.

Note | If the axis label is not displayed, then the EXTRACTSCALE=TRUE option is ignored.

Note | When EXTRACTSCALE=TRUE and a scale is extracted, the tick values are formatted to provide the best fit on the axis. In that case, the tick value format might differ from the data format even when a named format is applied to the data values.

See | “boolean” on page 1479 for other Boolean values that you can use.

EXTRACTSCALETYPE=DEFAULT | SCIENTIFIC
specifies whether to extract a named scale or a scientific-notation scale.

DEFAULT extracts a named scale.
A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as $10^n$) for values over 999 trillion. For large tick values, the scale factor is set to ensure that the absolute value of the largest value is greater than 1. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of $1/10$ ($10^{-n}$) for values less than 1 trillionth.

SCIENTIFIC extracts a scientific-notation scale.
A scientific-notation scale is a multiple of 10 expressed as $10^n$ for values greater than 1, or a multiple of $1/10$ expressed as $10^{-n}$ for values less than 1.

Default | DEFAULT

Restriction | The scale is derived from the English locale for all locales.

DATA uses the format that has been assigned to the column that is contributing to the axis (or BEST6 if no format is assigned) in order to control the formatting of the major tick values.

format specifies a format to apply to the major tick values.

Restriction | GTL currently honors most, but not every, SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

Note | If you specify a format that significantly reduces precision, because of tick-value rounding, the plot data elements might not
align properly with the axis tick values. In that case, specify a
tick-value format with a higher precision.

**Default**

(MAXWIDTH=8, MAXDECIMALS=6, PREFERREDDECIMALS=2,
EXTRACTSCALE=FALSE, EXTRACTSCALETYPE=DEFAULT)

**Interaction**

This option is ignored when the TICKDISPLAYLIST= option is
specified.

**TICKVALUETYPE=(numeric-list)**

specifies the tick values for a linear axis as a list.

**Default**

An internal algorithm determines the tick marks, based on the
actual axis data range or the data range established by the
VIEWMIN= and VIEWMAX= options. By default, when this option
is used, the only tick values that appear are the tick values in
numeric-list that fall within the explicit data range (set by
VIEWMIN= and VIEWMAX=) or the implicit data range (set by the
actual data minimum and data maximum).

**Restriction**

This option applies to linear axes only.

**Requirement**

The tick values must be specified as a space-separated list of
numeric values, enclosed in parentheses.

**Interactions**

This option overrides the INTEGER= option.

This option is ignored if the TICKVALUESEQUENCE= option is
specified, or if the DISPLAY= option or the
DISPLAYSECONDARY= option does not display tick values.

The VIEWMIN= and VIEWMAX= options alter the axis data range.
If the VIEWMIN= option is set to the minimum tick list value and the
VIEWMAX= option is set to the maximum tick list value, then all
ticks in the tick list are displayed. This might result in some data
not being displayed. For example, data might not be displayed
when the VIEWMIN= value is greater than the actual data
minimum, or when the VIEWMAX= value is less than actual data
maximum.

If TICKVALUEPRIORITY= TRUE, then the VIEWMIN= and
VIEWMAX= options are ignored if they are fully enclosed by the
numeric-list. The tick numeric-list can extend the implicit data
range of the axis, but cannot reduce it.

If MINORTICKS=TRUE is specified and TICKVALUELIST=
specifies nonuniform tick values such as 3, 5, 11, and 23, then
MINORTICK=TRUE is ignored. To display minor tick marks in that
case, specify uniform tick values such as 0, 10, 20, and 30 in the
TICKVALUELIST= option.

**Tip**

The values in the list are formatted according to the setting for the
TICKVALUEFORMAT= option.

**TICKVALUEPRIORITY=TRUE | FALSE**

specifies whether an axis tick specification (TICKVALUELIST= or
TICKVALUESEQUENCE=) can extend the axis data range.
**TRUE**
extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by either the `TICKVALUELIST=` or `TICKVALUESEQUENCE=` option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

**FALSE**
displays only the tick values that are specified by the `TICKVALUELIST=` option that fall within the explicit data range set by the `VIEWMIN=` and `VIEWMAX=` options or by the implicit data range set by the actual data minimum and data maximum.

Default | FALSE  
Restriction | This option applies to linear axes only.  
Interactions | When this option is set to TRUE, the `VIEWMIN=` and `VIEWMAX=` options are ignored.  
This option is ignored if the `TICKVALUELIST=` or `TICKVALUESEQUENCE=` option is not specified.  
This option is ignored if the `DISPLAY=` option or the `DISPLAYSECONDARY=` option does not display the tick values.  
Note | If the minimum and maximum of the specified values are within the data range, then this option has no effect.  
See | “boolean” on page 1479 for other Boolean values that you can use.

**TICKVALUEROTATION=**DIAGONAL | DIAGONAL2 | VERTICAL
specifies how the tick values are rotated on the X and X2 axes.

**DIAGONAL**
rotates the tick values to a 45-degree diagonal position.
The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

**DIAGONAL2**
rotates the tick values to a –45-degree diagonal position.
The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**VERTICAL**
rotates the labels to a 90-degree vertical position.
The labels are always drawn from bottom to top.

Default | DIAGONAL  
Restriction | This option is valid for `COLUMNAXISOPTS=` and `COLUMN2AXISOPTS=` only.
The `TICKVALUEFITPOLICY=` option must be set to `ROTATE` or `ROTATEALWAYS` for this option to have any effect.

**TICKVALUESEQUENCE=(START=number END=number INCREMENT=number)**

specifies the tick values by start, end, and increment.

**START=number**

specifies the value for the first tick mark.

**END=number**

specifies the value for the last tick mark.

**INCREMENT=number**

specifies the increment for intermediate tick marks between the first and last tick marks.

The END value always controls the last tick mark. The interval between the last tick mark and the previous tick mark might not necessarily be the INCREMENT value.

**Default**

An internal algorithm determines the tick marks, based on the actual axis data range or the data range established by the `VIEWMIN=` and `VIEWMAX=` options. By default, when this option is used, the only tick values that appear are those that fall within the explicit data range (set by `VIEWMIN=` and `VIEWMAX=`) or the implicit data range (set by the actual data minimum and data maximum).

**Interactions**

This option overrides the `INTEGER=` option.

The `VIEWMIN=` and `VIEWMAX=` options alter the axis data range. If the `VIEWMIN=` option is set to the START= option value and the `VIEWMAX=` option is set to the END= option value, then all ticks in the tick sequence are displayed.

If `TICKVALUEPRIORITY= TRUE`, then the tick sequence might extend the explicit data range of the axis, but never reduce it.

This option is ignored if the `DISPLAY=` option or the `DISPLAYSECONDARY=` option does not display tick marks.

**Tip**

The values in the sequence are formatted according to the setting for the `TICKVALUEFORMAT=` option.

**See**

`TICKVALUELIST=` option as an alternative for customizing tick marks.

**VIEWMAX=number**

specifies the maximum data value to include in the display. The value might be adjusted by the threshold calculation.

**Default**

The maximum value in the data for the specified axis.

**Interactions**

This option does not determine the maximum axis tick value that is displayed. The `THRESHOLDMAX=` value is used to determine the maximum tick value.

This option is ignored when `TICKVALUEPRIORITY= TRUE`. 

---

Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL 1185
### Notes

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

### Tip

To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

### See

"Adjusting the Axis View" on page 983

### VIEWMIN=\texttt{number}

Specifies the minimum data value to include in the display. The value might be adjusted by the threshold calculation.

**Default**

The minimum value in the data for the specified axis.

**Interactions**

This option does not determine the minimum axis tick value that is displayed. The THRESHOLDMIN= value is used to determine the minimum tick value.

This option is ignored when TICKVALUEPRIORITY= TRUE.

**Notes**

Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

### Options for Log Axes Only

The options that are documented in this section can be used with the \texttt{LOGOPTS=} axis option. The following table provides a summary of the options.

<table>
<thead>
<tr>
<th>Log Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{BASE}</td>
<td>Specifies the base of the logarithmic scale for the axis values.</td>
</tr>
<tr>
<td>\texttt{MINORGRID}</td>
<td>Specifies whether grid lines are displayed at the minor tick marks.</td>
</tr>
<tr>
<td>\texttt{MINORGRIDATTRS}</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>\texttt{MINORTICKCOUNT}</td>
<td>Specifies the number of minor ticks that are displayed on the axis.</td>
</tr>
</tbody>
</table>
### Log Axis Option

<table>
<thead>
<tr>
<th>Log Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>TICKINTERVALSTYLE</td>
<td>Specifies how to scale and format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT=</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the tick values for a log axis as a space-separated list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether the TICKVALUELIST specification can extend the axis data range.</td>
</tr>
<tr>
<td>VALUESTYPE</td>
<td>Specifies the scale that the system uses when interpreting the TICKVALUELIST=, VIEWMAX=, and VIEWMIN= option values.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display.</td>
</tr>
</tbody>
</table>

**BASE=10 | 2 | E**  
specifies the base of the logarithmic scale for the axis values.

- **Default**: 10  
- **Restriction**: This option applies to log axes only.

**MINORGRID=TRUE | FALSE**  
specifies whether grid lines are displayed at the minor tick marks.

![TRUE](image1.png)  
![FALSE](image2.png)

- **Defaults**: FALSE in SAS 9.4M1 and earlier releases.
- The GraphMinorGridLines:DisplayOpts style reference starting with SAS 9.4M2. If attribute DisplayOpts is not defined in the active style, then FALSE is the default value.
Interaction  This option is ignored if the `GRIDDISPLAY=` option does not display the grid lines.

Tips  The GRIDATTRS= option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the MINORGRIDATTRS= option.

Use the MINORTICKS= option to display the minor tick marks on the axis.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**MINORGRIDATTRS=**`style-element | style-element (line-options) | (line-options)`

specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

![Minor Grid Lines Example](image)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults  The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction  This option is ignored when MINORTICKS=FALSE.

Note  When `style-element` is specified, only the style element’s CONTRASTCOLOR, LIFESTYLE, and LINETHICKNESS attributes are used.

Tip  Use the `GRIDATTRS=` option to control the appearance of the major grid lines.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example  Here is an example that specifies light blue, dotted lines for the minor grid.

```plaintext
minorgridattrs=(color=lightblue pattern=dot);
```

**MINORTICKCOUNT=**`positive-integer`

specifies the number of minor ticks that are displayed on the axis.
Default

Eight ticks with nine intervals (BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT).

Restriction

Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

Interactions

The DISPLAY= or DISPLAYSECONDARY= option specification must include TICKS for this option to have any effect.

The MINORTICKS= option must specify TRUE for this option to have any effect.

Tip

To display \( n \) intervals between major ticks, use MINORTICKCOUNT=\( n-1 \).

MINORTICKS=TRUE | FALSE

specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

```
 TRUE
 1 10 100 1000

 FALSE
 1 10 100 1000
```

Default

FALSE

Restriction

Minor ticks can be displayed only when BASE=10 and TICKINTERVALSTYLE= is LOGEXPAND or LOGEXPONENT.

Requirement

If you set this option to TRUE, you must also specify a value for MINORTICKCOUNT=. Otherwise, minor tick marks do not appear on the log axis.

Interaction

If MINORTICKS=TRUE is specified and TICKVALUELIST= specifies nonuniform log tick values such as 10, 20, 30, and 40, then MINORTICK=TRUE is ignored. To display minor tick marks in that case, specify uniform log tick values such as 1, 10, 100, and 1000 in the TICKVALUELIST= option.

Tip

Use the MINORGRID= option to display grid lines at the minor tick values.

See

"boolean " on page 1479 for other Boolean values that you can use.

TICKINTERVALSTYLE=AUTO | LOGEXPAND | LOGEXPONENT | LINEAR

specifies how to scale and format the values for major tick marks.

AUTO

selects a LOGEXPAND, LOGEXPONENT, or LINEAR representation automatically based on the range of the data.

When the data range is small (within an order of magnitude), a LINEAR representation is typically used. Data ranges that encompass several orders of magnitude typically use the LOGEXPAND or LOGEXPONENT representation.
LOGEXPAND
places the major tick marks at uniform intervals at integer powers of the base.
The tick values are expanded as follows:

<table>
<thead>
<tr>
<th>Base=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 10 100 1000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Base=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 4 8 16 32 64 128 256 512 1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Base=E</th>
</tr>
</thead>
<tbody>
<tr>
<td>e^0 e^1 e^2 e^3 e^4 e^5 e^6 e^7</td>
</tr>
</tbody>
</table>

LOGEXPONENT
places the major tick marks at uniform intervals at integer powers of the base.
The tick values are only the integer exponents for all bases.

<table>
<thead>
<tr>
<th>Base=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3</td>
</tr>
</tbody>
</table>

LINEAR
places the major tick marks at non-uniform intervals that cover the range of the data.

<table>
<thead>
<tr>
<th>Base=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 40 200 400 600 1200</td>
</tr>
</tbody>
</table>

Default: AUTO

Restrictions
This option applies to log axes only.

For LOGEXPONENT, formats on data columns contributing to the axis are ignored. For LOGEXPAND, formats on data columns contributing to the axis are ignored, although any "named format" on the column is retained. For LINEAR, ticks values are automatically formatted when the column format is not assigned or one of w.d, Ew., or BESTw. Other formats (SAS defined or user-defined) are used if specified.

GTL currently honors most but not every SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

Note
When BASE=10 and LOGEXPAND or LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

Tip
If you use TICKINTERVALSTYLE=LOGEXPONENT, then you might want to include information in the axis label about which base is used.

TICKVALUEFORMAT=DATA | format
specifies how to format the values for major tick marks.

Note: This feature applies to SAS 9.4M3 and to later releases.
DATA uses the format that has been assigned to the column that is contributing to
the axis (or BEST6 if no format is assigned) in order to control the formatting
of the major tick values.

format specifies a format to apply to the major tick values.

Restriction GTL currently honors most, but not every, SAS format. For
details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

Restriction This option applies to log axes only.

Interactions This option is ignored when
TICKINTERVALSTYLE=LOGEXPONENT.

When TICKINTERVALSTYLE=LOGEXPAND, this option is honored
for the base 10 and base 2 logarithmic scales, and is ignored for the
base E scale.

When TICKINTERVALSTYLE=LINEAR, this option is honored for
the base 10, base 2, and base E logarithmic scales.

See

BASE=

TICKINTERVALSTYLE=

TICKVALUELIST=(numeric-list)
specifies the tick values for a linear axis as a list.

Default Only the tick values specified in the list that fall within the explicit
data range set by the VIEWMIN= and VIEWMAX= options or by
the implicit data range set by the actual data minimum and data
maximum are displayed. An internal algorithm determines the tick
marks.

Requirements The tick values must be specified as a space-separated list of
numeric values, enclosed in parentheses.

The values that you specify must be appropriate for the
VALUESTYPE= specification. Otherwise, unexpected results
might occur. If VALUESTYPE=EXPANDED is in effect (default),
specify increments of the log base power such as 0.1, 1, 10, 100,
and so on, on a base 10 log axis, for example. If
VALUESTYPE=EXPONENT is in effect, specify integer
increments of the log base power exponent such as 1, 2, 3, and
so on.

Interactions The VALUESTYPE= option determines how the values in the list
are interpreted.

The VIEWMIN= and VIEWMAX= options alter the axis data range.
If the VIEWMIN= option is set to the minimum tick list value and
the VIEWMAX= option is set to the maximum tick list value, then
all ticks in the tick list are displayed. This might result in some
data not being displayed. For example, data might not be
displayed when the VIEWMIN= value is greater than the actual
data minimum, or when the VIEWMAX= value is less than actual
data maximum.

If the VIEWMIN= value is greater than the actual data minimum or the VIEWMAX= value is less than actual data maximum, some
data might not be displayed.

This option is ignored if the DISPLAY= or the
DISPLAYSECONDARY= option does not display the tick values.

If MINORTICKS=TRUE is specified and TICKVALUELIST=
specifies nonuniform log tick values such as 10, 20, 30, and 40,
then MINORTICK=TRUE is ignored. To display minor tick marks
in that case, specify uniform log tick values such as 1, 10, 100,
and 1000 in the TICKVALUELIST= option.

See VIEWMIN= and VIEWMAX= options for controlling the data range
TICKINTERVALSTYLE= for specifying the scale and format of the
major tick values
TICKVALUEPRIORITY= for controlling the behavior of the
TICKVALUELIST= option
BASE= for specifying the log base

TICKVALUEPRIORITY=TRUE | FALSE
specifies whether the TICKVALUELIST= specification can extend the axis data
range.

TRUE
extends the axis data range (but does not reduce it) to include the minimum
and maximum values that are specified by the TICKVALUELIST= option.

If the minimum and maximum of the user-specified values are within the data
range, this option has no effect.

FALSE
displays only the tick values that are specified by the TICKVALUELIST=
option that fall within the explicit data range set by the VIEWMIN= and
VIEWMAX= options or by the implicit data range set by the actual data
minimum and data maximum.

Default FALSE

Interactions When this option is set to TRUE, the VIEWMIN= and VIEWMAX=
options are ignored.

This option is ignored if the DISPLAY= option or the
DISPLAYSECONDARY= option does not display the tick values.

This option is ignored if the TICKVALUELIST= option is not
specified.

Note If the minimum and maximum of the specified values are within
the data range, then this option has no effect.

See "boolean " on page 1479 for other Boolean values that you can use.
VALUETYPE=EXPANDED | EXPONENT

specifies the scale that the system uses when interpreting the TICKVALUELIST=, VIEWMAX=, and VIEWMIN= option values. Use this option to choose your preferred way of specifying log-axis values.

EXPANDED

values are interpreted as powers of the base such as 0.1, 1, 10, 100, and so on, for base 10, for example.

EXPONENT

values are interpreted as integer exponents of the base such as 1, 2, 3, and so on, for base 10, base 2, and base E.

Default EXPANDED

Note This option does not change the style of the tick values that are displayed on the axis. It changes only how the VIEWMIN=, VIEWMAX=, and TICKVALUELIST= option values are interpreted by the system.

Tip This option is particularly useful when BASE=E.

Examples

The following example specifies VIEWMIN= and VIEWMAX= as exponent values instead of as expanded values on an expanded Base 10 log axis. This results in X-axis tick values of 10, 100, 1000, 10000, and 100000.

```plaintext
xaxisopts=(type=log
  logopts=(base=10
    tickintervalstyle=logexpand
    valuestype=exponent
    viewmin=1 viewmax=5));
```

The following example specifies TICKVALUELIST= as a list of expanded values instead of exponent values on an exponent Base 10 log axis. This results in X-axis tick values of 1, 2, 3, 4, and 5.

```plaintext
xaxisopts=(type=log
  logopts=(base=10
    tickintervalstyle=logexponent
    tickvaluepriority=true
    valuestype=expanded
    tickvaluepriority=true
    viewmin=1 viewmax=5));
```

VIEWMAX=number

specifies the maximum data value to include in the display.

Default The maximum value in the data for the specified axis.

Requirement The value that you specify must be appropriate for the VALUETYPE= specification and the log base. Otherwise, unexpected results might occur. If VALUETYPE=EXPANDED is in effect (default), specify an increment of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If VALUETYPE=EXPONENT is in effect, specify an integer increment of the log base power exponent such as 1, 2, 3, and so on.

Interactions This option is ignored when TICKVALUEPRIORITY= TRUE.
The `VALUESTYPE=` option determines how the value is interpreted.

### Notes

Setting a `VIEWMAX=` or `VIEWMIN=` value does not alter the original data or any calculations on it.

If an invalid value is specified for the `VIEWMAX=` option, the default value for `VIEWMAX=` is used instead. In that case, if the default value for `VIEWMAX=` is less than the value specified by the `VIEWMIN=` option, then the `VIEWMIN=` and `VIEWMAX=` values are swapped.

The maximum axis tick value might differ from the `VIEWMAX=` value. The `VIEWMIN=` and `VIEWMAX=` values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

When `BASE=10` and `TICKINTERVALSTYLE=LOGEXPAND` or `TICKINTERVALSTYLE=LOGEXPONENT` is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

### Tip

To display the `VIEWMAX=` value as the maximum tick value, use the `TICKVALUELIST=` option.

### See

"Adjusting the Axis View" on page 983

### Examples

The following example specifies a value of 100,000 as an expanded value on a base 10 log axis:

```plaintext
VIEWMAX=100000
```

The following example specifies a value of 100,000 as an exponent value on a base 10 log axis:

```plaintext
VIEWMAX=5
```

### `VIEWMIN=number`

specifies the minimum data value to include in the display.

#### Default

The minimum value in the data for the specified axis.

#### Requirement

The value that you specify must be appropriate for the `VALUESTYPE=` specification and the log base. Otherwise, unexpected results might occur. If `VALUESTYPE=EXPANDED` is in effect (default), specify an increment of the log base power such as 0.1, 1, 10, 100, and so on, on a base 10 log axis, for example. If `VALUESTYPE=EXPONENT` is in effect, specify an integer increment of the log base power exponent such as 1, 2, 3, and so on.

#### Interactions

This option is ignored when `TICKVALUEPRIORITY=TRUE`.

#### Notes

Setting a `VIEWMAX=` or `VIEWMIN=` value does not alter the original data or any calculations on it.
The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

When BASE=10 and TICKINTERVALSTYLE=LOGEXPAND or TICKINTERVALSTYLE=LOGEXPONENT is used, an intermediate tick is added whenever the axis data range is less than or equal to 1.5 powers of 10.

**Tip**
To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

**See**
"Adjusting the Axis View" on page 983

**Examples**
The following example specifies a value of 0.1 as an expanded value on a base 10 log axis:

```
VIEWMIN=0.1
```

The following example specifies a value of 0.1 as an exponent value on a base 10 log axis:

```
VIEWMIN=-1
```

### Options for Time Axes Only

The options that are documented in this section can be used with the TIMEOPTS= axis option. Some of the options are axis-specific. The following tables provide a summary of the time-axes options based on the layout axis option in which it is used.

**Table 8.4  Time-Axis Options for the COLUMNAXIS= or COLUMN2AXIS=Layout Option**

<table>
<thead>
<tr>
<th>Time Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
<td>Specifies that evenly spaced integer values are used for tick marks.</td>
</tr>
<tr>
<td>INTERVALMULTIPLIER</td>
<td>Specifies a multiplier to apply to the time interval that is in effect for the axis.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKINTERVAL</td>
<td>Specifies the time interval between minor ticks.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>SPLITTIICKVALUE</td>
<td>Specifies whether to split the tick values on an X or X2 axis, if possible.</td>
</tr>
<tr>
<td>TICKVALUEFITPOLICY</td>
<td>Specifies a policy for avoiding tick value collision on an X or X2 axis.</td>
</tr>
</tbody>
</table>
## Table 8.5  Time-Axis Options for theROWAXIS= or ROW2AXIS=Layout Option

<table>
<thead>
<tr>
<th>Time Axis Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTERVAL</td>
<td>Specifies that evenly spaced integer values are used for tick marks.</td>
</tr>
<tr>
<td>INTERVALMULTIPLIER</td>
<td>Specifies a multiplier to apply to the time interval that is in effect for the axis.</td>
</tr>
<tr>
<td>MINORGRID</td>
<td>Specifies whether grid lines are displayed at the minor tick values.</td>
</tr>
<tr>
<td>MINORGRIDATTRS</td>
<td>Specifies the attributes of the minor grid lines.</td>
</tr>
<tr>
<td>MINORTICKINTERVAL</td>
<td>Specifies the time interval between minor ticks.</td>
</tr>
<tr>
<td>MINORTICKS</td>
<td>Specifies whether minor ticks are displayed.</td>
</tr>
<tr>
<td>TICKVALUEFORMAT</td>
<td>Specifies how to format the values for major tick marks.</td>
</tr>
<tr>
<td>TICKVALUELIST</td>
<td>Specifies the order of the tick values for a time axis as list.</td>
</tr>
<tr>
<td>TICKVALUEPRIORITY</td>
<td>Specifies whether an axis tick specification that is specified with the TICKVALUELIST= option can extend the axis data range.</td>
</tr>
<tr>
<td>VIEWMAX</td>
<td>Specifies the maximum data value to include in the display.</td>
</tr>
<tr>
<td>VIEWMIN</td>
<td>Specifies the minimum data value to include in the display.</td>
</tr>
</tbody>
</table>
## Table 8.6  Time Intervals

<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>Unit</th>
<th>Tick interval</th>
<th>Default tick value format</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>DATE, TIME, or DATETIME</td>
<td>automatically chosen</td>
<td>automatically chosen</td>
</tr>
<tr>
<td>SECOND</td>
<td>TIME or DATETIME</td>
<td>second</td>
<td>TIME8.</td>
</tr>
<tr>
<td>MINUTE</td>
<td>TIME or DATETIME</td>
<td>minute</td>
<td>TIME8.</td>
</tr>
<tr>
<td>HOUR</td>
<td>TIME or DATETIME</td>
<td>hour</td>
<td>TIME8.</td>
</tr>
<tr>
<td>DAY</td>
<td>DATE or DATETIME</td>
<td>day</td>
<td>DATE9.</td>
</tr>
<tr>
<td>TENDAY</td>
<td>DATE or DATETIME</td>
<td>10 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>WEEK</td>
<td>DATE or DATETIME</td>
<td>7 days</td>
<td>DATE9.</td>
</tr>
<tr>
<td>SEMIMONTH</td>
<td>DATE or DATETIME</td>
<td>1st and 16th of each month</td>
<td>DATE9.</td>
</tr>
<tr>
<td>MONTH</td>
<td>DATE or DATETIME</td>
<td>month</td>
<td>MONYY7.</td>
</tr>
<tr>
<td>QUARTER</td>
<td>DATE or DATETIME</td>
<td>3 months</td>
<td>YYQC6.</td>
</tr>
<tr>
<td>SEMIYEAR</td>
<td>DATE or DATETIME</td>
<td>6 months</td>
<td>MONYY7.</td>
</tr>
<tr>
<td>YEAR</td>
<td>DATE or DATETIME</td>
<td>year</td>
<td>YEAR4.</td>
</tr>
</tbody>
</table>

**Default**  AUTO. An appropriate interval is chosen based on the data and the column date, date-time, or time format.

**Restriction**  This option applies to time axes only.

**Requirement**  The data column(s) mapped to a time axis must be in the same duration units: TIME, DATE, or DATETIME. The selection of an
interval must be consistent with the duration unit. For example, if the data are in time units, you can specify only AUTO, SECOND, MINUTE, HOUR.

**INTERVALMULTIPLIER=** *positive-integer*

specifies a multiplier to apply to the time interval that is in effect for the axis.

**Note:** This feature applies to **SAS 9.4M3** and to later releases.

**Default** 1

**Restriction** This option applies to time axes only.

**Interaction** This option is ignored if the **TICKVALUELIST=** option is used.

**Tip** Use the **INTERVAL=** option to specify a different time interval.

**Examples**

To specify 3-month intervals:

```
INTERVAL=MONTH INTERVALMULTIPLIER=3
```

To specify 10-year intervals:

```
INTERVAL=YEAR INTERVALMULTIPLIER=10
```

**MINORGRID=** **TRUE | FALSE**

specifies whether grid lines are displayed at the minor tick marks.

**Defaults** FALSE in **SAS 9.4M1** and earlier releases.

The **GraphMinorGridLines:DisplayOpts** style reference starting with **SAS 9.4M2**. If attribute **DisplayOpts** is not defined in the active style, then FALSE is the default value.

**Interaction** This option is ignored if the **GRIDDISPLAY=** option does not display the grid lines.

**Tips** The **GRIDATTRS=** option does not affect the appearance of the minor grid lines. To control the minor grid line appearance, use the **MINORGRIDATTRS=** option.

Use the **MINORTICKS=** option to display the minor tick marks on the axis.
MINORGRIDATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the minor grid lines. This option does not affect the major grid lines.

The following figure shows the minor grid lines set to light blue, dotted lines on a linear axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a base-10 log axis. (See the example.)

The following figure shows the minor grid lines set to light blue, dotted lines on a time axis. (See the example.)

Defaults
The GraphGridLines style element is used starting with SAS 9.4.

The GraphMinorGridLines style element is used starting with SAS 9.4M2.

Interaction
This option is ignored when MINORTICKS=FALSE.

Note
When style-element is specified, only the style element’s CONTRASTCOLOR, LINESTYLE, and LINETHICKNESS attributes are used.

Tip
Use the GRIDATTRS= option to control the appearance of the major grid lines.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style element.

“Line Options” on page 1522 for available line options.

Example
Here is an example that specifies light blue, dotted lines for the minor grid.

```
minorgridattrs=(color=lightblue pattern=dot);
```

MINORTICKINTERVAL=interval
specifies the time interval between minor ticks. See Table 8.13 on page 1197 for information about the intervals that you can select. The interval that you select must be consistent with the axis data duration units such as TIME, DATE, or DATETIME. For example, if the axis data is in TIME units, then you must specify AUTO, SECOND, MINUTE, or HOUR.

Default
AUTO

Interactions
This option is ignored if the TICKVALUELIST= option is used.

This option is ignored if the MINORTICKINTERVAL= setting is greater than the INTERVAL= setting.
MINORTICKS=TRUE | FALSE  
specifies whether minor ticks are displayed. When MINORTICKS=TRUE, the minor tick marks are displayed on the axis as shown in the following figure.

<table>
<thead>
<tr>
<th>TRUE</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>Feb</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Default: FALSE

Interactions: The number of minor ticks is dependent on the value of the MINORTICKINTERVAL= option, if specified. If MINORTICKINTERVAL= is not specified, then it is dependent on the value of the INTERVAL= option.

This option is ignored if the TICKVALUELIST= option is used, or if the DISPLAY= or DISPLAYSECONDARY= option does not display the tick marks.

If MINORTICKS=TRUE is specified with the TICKVALUELIST= option, then MINORTICKS=TRUE is ignored, and minor tick marks are not displayed.

Tip: Use the MINORGRID= option to display grid lines at the minor tick values.

See: “boolean” on page 1479 for other Boolean values that you can use.

SPLITTICKVALUE=TRUE | FALSE  
specifies whether to split the tick values on an X or X2 axis, if possible. This option is not available for a Y or Y2 axis.

TRUE  
splits the axis tick values into two lines allowing more tick values to appear.

For example, with INTERVAL= MONTH, this is how tick values are split:

|----------|----------|----------|----------|----------|----------|----------|----------|----------|

FALSE  
does not split the axis tick values.

For example, when this option specifies FALSE, this is how the tick values in the previous example appear:

|----------|----------|----------|----------|----------|----------|----------|

Typically, fewer tick values fit, causing thinning, rotation, or staggering of the values. See the TICKVALUEFITPOLICY= option.

Default: TRUE

Restrictions: This option applies to time axes only.
This option is valid only in the COLUMNAXIS= and COLUMN2AXIS= layout axis options.

Interaction
This option is ignored if the TICKVALUELIST= or TICKVALUEFORMAT= option is used.

See “boolean ” on page 1479 for other Boolean values that you can use.

**TICKVALUEFITPOLICY=** *policy*

specifies a policy for avoiding tick value collision on an X or X2 axis. This option is not available for the Y and Y2 axes. The effectiveness of a collision-avoidance policy depends on the number of tick values, their length, and the length of the axis. The following policies are valid:

**EXTRACT**

displays consecutive integers along the axis instead of the actual tick values in order to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement
The **EXTRACT** policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*.

**EXTRACTALWAYS**

same as **EXTRACT**, except that the extraction is implemented regardless of whether collision occurs.

Requirement
The **EXTRACTALWAYS** policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in *SAS Graph Template Language: User’s Guide*.

**NONE**

makes no attempt to avoid collisions between tick values.

Tick values are displayed even when they collide.

**SPLIT**

splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.

A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See **TICKVALUESPLITCHAR=**

**SPLITALWAYS**

always splits the axis tick value at every occurrence of a split character that is specified by the **TICKVALUESPLITCHAR=** option.

See **TICKVALUESPLITCHAR=**
SPLITALWAYSTHIN
same as SPLITALWAYS, except that thinning is performed when long words do not fit the available space.

SPLITTHIN
same as SPLIT, except that thinning is performed when long words do not fit the available space.

THIN
eliminates alternate tick values.

EXTRACT
display consecutive integers along the axis instead of the actual tick values to represent those tick values.

In most cases, this policy is implemented if the system estimates that a collision might occur. If no collision occurs, then the actual tick values are displayed on the axis in the normal manner.

Requirement The EXTRACT policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

EXTRACTALWAYS
same as EXTRACT, except that the extraction is implemented regardless of whether collision occurs.

Requirement The EXTRACTALWAYS policy must be used with an AXISLEGEND statement. For more information, see “Extracting Discrete Axis Tick Values into a Legend” in SAS Graph Template Language: User’s Guide.

NONE
does not attempt to fit tick values that collide.

ROTATE
rotates the tick values if a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYS
rotates the tick values regardless of whether a collision occurs.

The TICKVALUEROTATION= option specifies whether the values are rotated to a 45-degree diagonal or a 90-degree vertical position. By default, the values are rotated to a 45-degree diagonal position.

ROTATEALWAYSDROP
attempts the ROTATEALWAYS policy, and then drops the tick values if collisions still occur.

ROTATETHIN
attempts the ROTATE policy, and then the THIN policy.

SPLIT
splits the tick value at a split character, which is specified by the TICKVALUESPLITCHAR= option, only when necessary in order to make the value fit the available space.
A split does not occur at a split character if a split is not needed at that location. If the value does not contain any of the specified split characters, then the value is not split. Values that are not split or that do not fit the available space even after splitting might overlap the adjoining space.

See **TICKVALUESPLITCHAR**=

**SPLITALWAYS**
always splits the axis tick value at every occurrence of a split character that is specified by the **TICKVALUESPLITCHAR**= option.

See **TICKVALUESPLITCHAR**=

**SPLITROTATE**
attempts the SPLIT policy, and then the ROTATE policy.

**STACKEDALWAYS**
always displays the tick values vertically as stacked letters.

---

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**STACKEDALWAYSTHIN**
always displays the tick values vertically as stacked letters.
Thinning is performed when the tick values do not fit the available space.

---

**Note:** This feature applies to SAS 9.4M5 and to later releases.

**STAGGER**
alternates the tick values between two rows.

**STAGGERROTATE**
attempts the STAGGER policy, and then the ROTATE policy.

**STAGGERTHIN**
attempts the STAGGER policy, and then the THIN policy.

**STAGGERTRUNCATE**
attempts the STAGGER policy, and then the TRUNCATE policy.

**THIN**
eliminates alternate tick values.

**TRUNCATE**
shortens the tick values when they exceed a certain number of characters.

**TRUNCATEROTATE**
attempts the TRUNCATE policy, and then the ROTATE policy.

**TRUNCATESTAGGER**
attempts the TRUNCATE policy, and then the STAGGER policy.

**TRUNCATETHIN**
attempts the TRUNCATE policy, and then the THIN policy.

**Default**
**THIN**

**Restriction**
This option is valid only for the X and X2 axes.
When `SPLITTICKVALUE= TRUE`, this option is ignored and only the THIN policy is used.

A note is written to the SAS log when tick value thinning occurs.

**TICKVALUEFORMAT=** *format* | *DATA*

specifies how to format the values for major tick marks.

*format*

specifies a SAS date, time, or datetime format to control how the major tick values are displayed.

This format must be in the same duration units as the data column(s) mapped to a time axis: TIME, DATE, or DATETIME and should be appropriate for the value of the `INTERVAL=` option. For example, if `INTERVAL=MONTH` and there are two years of data displayed on the axis, then choosing `TICKVALUEFORMAT=YEAR.` would result in several ticks having the same year value.

*DATA*

specifies that the SAS date, time, or datetime format associated with the data column assigned to the axis be used to control how the major tick values are displayed.

**Default**
The default format used by the `INTERVAL=` option. The default does not apply if `TICKVALUELIST=` is specified.

**Restrictions**
This option applies to time axes only.

GTL currently honors most but not every SAS format. For details, see Appendix 5, “SAS Formats Not Supported,” on page 1541.

**TICKVALUELIST=(time-constant-list | date-constant-list | datetime-constant-list | numeric-list)**

specifies the tick values for a time axis as list.

**Default**
An internal algorithm determines the tick values.

**Restrictions**
This option applies to time axes only.

If `TICKVALUEPRIORITY=` is set to FALSE, then this option does not extend the data range of the axis. If the values fall within the default data range or that specified by the `VIEWMIN=` or `VIEWMAX=` options, then they are used.

**Requirement**
The tick values must be specified as a space-separated list of values enclosed in parentheses. The items in the list must be in the same duration units as the data mapped to the axis: TIME, DATE, or DATETIME. The values can be expressed as SAS TIME, DATE, or DATETIME constants (for example, "13:23"T, "11MAY06"D, or "11MAY06:13:23"DT) or their numeric equivalents.

**Interactions**
The values in the list are formatted according to the format specified on the `TICKVALUEFORMAT=` option. If `TICKVALUEFORMAT=` is not used, then the values are formatted according to the column format (the default `TICKVALUEFORMAT` value is not applied to these values).
If this option is specified, the \texttt{SPLITTICKVALUE=} and \texttt{INTERVAL=} options are ignored.

If \texttt{MINORTICKS=}\texttt{TRUE} is specified with the \texttt{TICKVALUELIST=} option, \texttt{MINORTICKS=}\texttt{TRUE} is ignored, and minor tick marks are not displayed.

\textbf{TICKVALUEPRIORITY=}\texttt{TRUE | FALSE}

specifies whether the \texttt{TICKVALUELIST=} specification can extend the axis data range.

\textbf{TRUE}

extends the axis data range (but does not reduce it) to include the minimum and maximum values that are specified by the \texttt{TICKVALUELIST=} option.

If the minimum and maximum of the user-specified values are within the data range, this option has no effect.

\textbf{FALSE}

displays only the tick values that are specified by the \texttt{TICKVALUELIST=} option that fall within the explicit data range set by the \texttt{VIEWMIN=} and \texttt{VIEWMAX=} options or by the implicit data range set by the actual data minimum and data maximum.

\textbf{Default} \hspace{1cm} \texttt{FALSE}

\textbf{Restriction} \hspace{1cm} This option is valid only in the \texttt{ROWAXIS=} and \texttt{ROW2AXIS=} layout axis options.

\textbf{Interactions} \hspace{1cm} When this option is set to \texttt{TRUE}, the \texttt{VIEWMIN=} and \texttt{VIEWMAX=} options are ignored.

This option is ignored if the \texttt{DISPLAY=} option or the \texttt{DISPLAYSECONDARY=} option does not display the tick values.

This option is ignored if the \texttt{TICKVALUELIST=} option is not specified.

\textbf{Note} \hspace{1cm} If the minimum and maximum of the specified values are within the data range, then this option has no effect.

\textbf{See} \hspace{1cm} "boolean " on page 1479 for other Boolean values that you can use.

\textbf{TICKVALUEROTATION=}\texttt{DIAGONAL | DIAGONAL2 | VERTICAL}

specifies how the tick values are rotated on the X and X2 axes.

\textbf{DIAGONAL}

rotates the tick values to a 45-degree diagonal position.

The X labels read left to right in a downward direction. The X2 labels read left to right in an upward direction.

\textbf{DIAGONAL2}

rotates the tick values to a –45-degree diagonal position.

The X labels read left to right in an upward direction. The X2 labels read left to right in a downward direction.

\textbf{Note:} This feature applies to \texttt{SAS 9.4M5} and to later releases.
VERTICAL
rotates the labels to a 90-degree vertical position.
The labels are always drawn from bottom to top.

Default: DIAGONAL

Restriction: This option is valid for COLUMN1AXISOPTS= and COLUMN2AXISOPTS= only.

Interaction: The TICKVALUEFITPOLICY= option must be set to ROTATE or ROTATEALWAYS for this option to have any effect.

**VIEWMAX=** *number*
specifies the maximum data value to include in the display.

Default: The maximum value in the data for the specified axis.

Notes: Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The maximum axis tick value might differ from the VIEWMAX= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip: To display the VIEWMAX= value as the maximum tick value, use the TICKVALUELIST= option.

See: “Adjusting the Axis View” on page 983

**VIEWMIN=** *number*
specifies the minimum data value to include in the display.

Default: The minimum value in the data for the specified axis.

Notes: Setting a VIEWMAX= or VIEWMIN= value does not alter the original data or any calculations on it.

The minimum axis tick value might differ from the VIEWMIN= value. The VIEWMIN= and VIEWMAX= values, and additional factors such as thresholds and the tick values computed by the plot statement, are used to determine the axis tick values.

Tip: To display the VIEWMIN= value as the minimum tick value, use the TICKVALUELIST= option.

See: “Adjusting the Axis View” on page 983

**Details**

The `LAYOUT DATALATTICE` and `LAYOUT DATAPANEL` statements each create a grid of graphs based on the values of one or more classification variables. In the grid, the axes are always external and displayed on the “primary” axes by default. The axes that are considered primary depend on the settings for the XAXIS= and YAXIS= options in plot statements that are specified within the LAYOUT PROTOTYPE. For managing the primary axes, both the `LAYOUT DATALATTICE` and `LAYOUT DATAPANEL` statements have COLUMNAXISOPTS=,
COLUMN2AXISOPTS=, ROWAXISOPTS=, and ROW2AXISOPTS= options that manage the axis features separately for columns and rows. The settings that are available can manage odd and even columns and rows separately, enabling you to simplify the axis display within the grid.

The following table shows which axis is primary for the XAXIS= and YAXIS= settings, and which axis option to use to manage that primary axis.

<table>
<thead>
<tr>
<th>Option Setting</th>
<th>Primary Axis</th>
<th>Axis Option to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAXIS=X</td>
<td>X (bottom)</td>
<td>COLUMNAXISOPTS=</td>
</tr>
<tr>
<td>XAXIS=X2</td>
<td>X2 (top)</td>
<td>COLUMN2AXISOPTS=</td>
</tr>
<tr>
<td>YAXIS=Y</td>
<td>Y (left)</td>
<td>ROWAXISOPTS=</td>
</tr>
<tr>
<td>YAXIS=Y2</td>
<td>Y2 (right)</td>
<td>ROW2AXISOPTS=</td>
</tr>
</tbody>
</table>

The settings that are available for the axis options can manage odd and even columns and rows separately, enabling you to simplify the axis display within the grid.

- To manage the first, third, and odd occurrences of a primary axis, use the DISPLAY= option.
- To manage the second, fourth, and even occurrences of a primary axis, use the ALTDISPLAY= option.

You can also display “secondary” axes in the grid. A secondary axis is not an independent axis. Rather, it mirrors the primary axis, but it is displayed on the opposite side and can have different display options. For example, when the X axis (bottom) is primary, you can mirror that axis with a secondary X axis at the top of the grid. Similarly, when the Y2 axis (right) is primary, you can mirror that axis with a secondary Y2 axis on the left of the grid. A secondary axis makes it easier to interpolate values in the cells that are farthest away from the primary axis.

Secondary axes can be displayed in the graph, provided all plot statements in the LAYOUT PROTOTYPE map data to the same primary axis. For example, a secondary X axis can be displayed at the top of the layout, provided all plot statements set XAXIS=X to map data to the primary X axis (bottom). Similarly, a secondary Y2 axis can be displayed to the left in the layout, provided all plot statements set YAXIS=Y2 to map data to the primary Y2 axis (right). If all plot statements in the LAYOUT PROTOTYPE do not map data to the same primary axis, then the secondary axes are not displayed.

To display secondary axes in the grid, use the DISPLAYSECONDARY= and ALTDISPLAYSECONDARY= options. As with the options for the primary axes, the DISPLAYSECONDARY= option manages the first, third, and odd occurrences of a secondary axis. The ALTDISPLAYSECONDARY= option manages the second, fourth, and even occurrences of a secondary axis.

In the default cases for the plots within the layout, the axis type is always DISCRETE, LINEAR, or TIME. The TYPE= option enables you to specify an axis type that overrides the default. For example, when appropriate for the data, you can request a LOG axis. When you override the default axis type, you must be sure to specify the correct axis type for the plot(s) that you are defining.
Each axis type has features specific to that type, and the following axis options enable you to specify features for the different types: `DISCRETEOPTS=`, `LINEAROPTS=`, `LOGOPTS=`, and `TIMEOPTS=`. One or more of these options can be specified for an axis, but the specified settings are applied only to the axis type that supports them.

**Note:** Certain plot types or layouts sometimes impose restrictions on what type of axis can be assigned. See the plot or layout documentation for default axis types and any restrictions that might apply.

---

**Example: Axis Options for LAYOUT DATALATTICE and LAYOUT DATAPANEL**

The following graph was generated by the "Example Program" on page 1208:

### Annual Furniture Sales Comparisons

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Canada</td>
<td>$80,000</td>
<td>$50,000</td>
<td>$55,000</td>
<td>$45,000</td>
</tr>
<tr>
<td>Mexico</td>
<td>$250,000</td>
<td>$225,000</td>
<td>$200,000</td>
<td>$175,000</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>$175,000</td>
<td>$150,000</td>
<td>$125,000</td>
<td>$100,000</td>
</tr>
</tbody>
</table>

---

**Example Program**

This example shows how axis attributes can be managed separately for even and odd columns and rows in the layout grid. In this case, the `ROWAXISOPTS=` option is used to stagger the Y-axes attributes. On the primary (left) Y axis, `DISPLAY=` displays TICKS and TICKVALUES on the first and third rows while `ALTDISPLAY=` displays just TICKS on the second row. On the secondary (right) Y axis, `DISPLAYSECONDARY=` displays just TICKS on the first and third rows, while `ALTDISPLAYSECONDARY=` displays TICKS and TICKVALUES on the second row. This alternating pattern could also have been set for the column axes. The pattern is independent of the number of rows and columns.

```plaintext
proc template;
    define statgraph layoutdatalattice;
```

---
begingraph;
    entrytitle "Annual Furniture Sales Comparisons"
    layout datalattice rowvar=country columnvar=year / rowdatarange=union
    headerlabellocation=inside
    headerlabeldisplay=value
    headerbackgroundcolor=GraphAltBlock:color
    rowaxisopts=(griddisplay=on
display=(tickvalues)
altdisplay=(ticks)
displaysecondary=(ticks)
altdisplaysecondary=(ticks tickvalues)
linearopts=(tickvalueformat=dollar12.))
columnaxisopts=(display=(tickvalues)
timeopts=(tickvalueformat=monname3));
    layout prototype / cycleattrs=true;
    seriesplot x=month y=TotalActual / name="Actual"
    seriesplot x=month y=TotalPredict / name="Predict"
    endlayout;
    sidebar / align=bottom;
    discretelegend "Actual" "Predict" / border=false;
    endsidebar;
endlayout;
endgraph;
end;

proc summary data=sashelp.prdsal2 nway;
    class country year month;
    var actual predict;
    output out=prdsal2 sum=TotalActual TotalPredict;
run;

proc sgrender data=prdsal2 template=layoutdatalattice;
run;
PART 6
Legend Statements

Chapter 9
Legend Statements .................................................. 1213
Legend Statements

Dictionary

AXISLEGEND Statement
Generates consecutive integers for display as axis-tick values in the graph, and creates a legend that correlates those integers with the actual tick values that they represent.

Requirements:
The axis must be named with the axis NAME= option.
The axis must be a discrete axis (TYPE=DISCRETE).
The axis must use the TICKVALUEFITPOLICY=EXTRACT or TICKVALUEFITPOLICY=EXTRACTALWAYS suboption of the DISCRETEOPTS= axis option.

Interactions:
If the total legend area in the graph exceeds the percentage that is set by the MAXLEGENDAREA= option in an ODS GRAPHICS statement that is in effect for the output destination, the legend might be dropped.
If DISPLAYCLIPPED=FALSE and the full legend cannot be displayed, the legend might be dropped.

Syntax
AXISLEGEND "axis-name" <options>;

Summary of Optional Arguments

Appearance options
ACROSS=positive-integer
specifies the number of legend entries that are placed horizontally before the next row begins.
BACKGROUND_COLOR = \texttt{style-reference} | \texttt{color}

specifies the color of the legend background.

\texttt{BORDER} = \texttt{TRUE} | \texttt{FALSE}

specifies whether a border is displayed around the legend.

\texttt{BORDERATTRS} = \texttt{style-element} \texttt{| style-element (line-options)} | \texttt{(line-options)}

specifies the attributes of the border line around the legend.

\texttt{DISPLAYCLIPPED} = \texttt{TRUE} | \texttt{FALSE}

specifies whether the legend is displayed when any portion of the legend cannot be rendered entirely.

\texttt{DOWN} = \texttt{positive-integer}

specifies the number of legend entries that are placed vertically before the next column begins.

\texttt{OPAQUE} = \texttt{TRUE} | \texttt{FALSE}

specifies whether the legend background is opaque (TRUE) or transparent (FALSE).

\texttt{ORDER} = \texttt{ROWMAJOR} | \texttt{COLUMNMAJOR}

specifies whether legend entries are organized into rows or into columns.

\texttt{OUTERPAD} = \texttt{AUTO} | \texttt{dimension} \texttt{| (pad-options)}

specifies the amount of extra space that is added outside the legend border.

\texttt{PAD} = \texttt{dimension} \texttt{| (pad-options)}

specifies the amount of extra space that is reserved inside the legend perimeter.

\texttt{VALUEATTRS} = \texttt{style-element} \texttt{| style-element (text-options)} | \texttt{(text-options)}

specifies the color and font attributes of the legend values.

Location options

\texttt{AUTOALIGN} = \texttt{NONE} | \texttt{AUTO} \texttt{| (location-list)}

specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.

\texttt{HALIGN} = \texttt{CENTER} | \texttt{LEFT} | \texttt{RIGHT} \texttt{| number}

specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

\texttt{LOCATION} = \texttt{OUTSIDE} | \texttt{INSIDE}

specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or REGION layout.

\texttt{VALIGN} = \texttt{CENTER} | \texttt{TOP} | \texttt{BOTTOM} \texttt{| number}

specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

Text options

\texttt{TITLE} = \texttt{"string"}

specifies the title of the legend.

\texttt{TITLEATTRS} = \texttt{style-element} \texttt{| style-element (text-options)} | \texttt{(text-options)}

specifies the color and font attributes of the legend title.

Required Argument

\texttt{"axis-name"}

references an axis’s name. The name controls which axis determines the legend entries.
Requirement

The axis name must be enclosed in quotation marks.

Optional Arguments

### ACROSS=positive-integer

Specifies the number of legend entries that are placed horizontally before the next row begins.

**Default**

The entries are placed to best fit the available area. This “best fit” approach works only when the legend is nested in the template’s outermost layout.

**Interaction**

This option is ignored if ORDER= COLUMNMAJOR

### AUTOALIGN=NULL | AUTO | (location-list)

Specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.

**NONE**

Does not automatically align the legend within its parent layout.

The legend’s position is set by the HALIGN= and VALIGN= options.

**AUTO**

Within the parent layout, attempts to center the legend in the area that is farthest from any surrounding data point markers.

**(location-list)**

Within the parent layout, restricts the legend’s possible locations to those locations in the specified location-list, and uses the location-list position that least collides with the parent layout’s other graphics features.

The location-list is space-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

**Default**

NONE

**Restriction**

AUTO is available only when the parent layout contains a scatter plot. Otherwise, it is ignored.

**Interactions**

This option has no effect unless LOCATION= INSIDE.

When LOCATION=INSIDE and AUTOALIGN= is not NONE, the HALIGN= and VALIGN= options are ignored.

**See**

LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type layout.

### BACKGROUNDCOLOR=style-reference | color

Specifies the color of the legend background.

**style-reference**

Specifies a style reference in the form style-element:style-attribute.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

**Default**

The GraphLegendBackground:Color style reference.
Interaction  OPAQUE= TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

**BORDER=TRUE | FALSE**
specifies whether a border is displayed around the legend.

Defaults  TRUE in SAS 9.4M1 and earlier releases.

The GraphLegendBackground:FrameBorder style reference starting with SAS 9.4M2. If attribute FrameBorder is not defined in the active style, then TRUE is the default value.

Tip  The BORDERATTRS= option controls the appearance of the legend border.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=style-element | style-element (line-options) | (line-options)**
specifies the attributes of the border line around the legend.

Default  The GraphBorderLines style element.

Interaction  BORDER= TRUE must be set for this option to have any effect.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element

“Line Options” on page 1522 for available line-options.

**DISPLAYCLIPPED=TRUE | FALSE**
specifies whether the legend is displayed when any portion of the legend cannot be rendered entirely. Based on the legend contents and placement, or when the graph size is reduced, parts of the legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and this option is set to FALSE, the entire legend is removed from the graph. The space for that legend is then reclaimed by the remainder of the graph. When this option is set to TRUE, the legend always appears, even if some parts of it have been clipped.

Default  FALSE

See  “boolean ” on page 1479 for other Boolean values that you can use.

**DOWN=positive-integer**
specifies the number of legend entries that are placed vertically before the next column begins.

Default  The entries are placed to best fit the available area. This “best fit” approach works only when the legend is nested in the template’s outermost layout.

Restriction  This option is ignored if ORDER= ROWMAJOR

**HALIGN=** CENTER | LEFT | RIGHT | number
specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

**number**
specifies an explicit position in the containing layout.
Range  The *number* specification can range from 0 to 1. The *number* represents a fraction of the parent container’s width, where 0 is all the way to the left and 1 is all the way to the right.

Interaction  For a *number* setting to take effect, LOCATION=INSIDE must be set. A *number* setting is invalid on this option when LOCATION=OUTSIDE.

Default  CENTER

Restriction  This option is available only when this statement is nested within an overlay-type or REGION layout.

Interactions  If LOCATION= OUTSIDE, then the HALIGN= and VALIGN= options cannot both be set to CENTER.

This option is ignored when LOCATION=INSIDE, the AUTOALIGN= option is not NONE, and the parent layout is an overlay-type layout.

See  LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

**LOCATION=OUTSIDE | INSIDE**  specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or REGION layout.

Default  OUTSIDE

Restriction  This option has effect only when the legend statement appears within an overlay-type or REGION layout and at least one stand-alone plot statement is referenced by the parent layout.

Interactions  The actual position is determined by the settings for the LOCATION=, AUTOALIGN=, HALIGN=, and VALIGN= options.

If this option is set to OUTSIDE, then the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The *number* setting for the alignment is invalid when the legend is positioned outside of the plot area.

See  LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

**OPAQUE=TRUE | FALSE**  specifies whether the legend background is opaque (TRUE) or transparent (FALSE).

Default  FALSE

Interaction  When this option is set to FALSE, the background color is not used.

See  “boolean ” on page 1479 for other Boolean values that you can use.

**ORDER=ROWMAJOR | COLUMNMAJOR**  specifies whether legend entries are organized into rows or into columns.

Default  ROWMAJOR
If ORDER=ROWMAJOR, then use the ACROSS= option to limit the number of entries in a row. If ORDER=COLUMNMAJOR, then use the DOWN= option to limit the number of entries in a column.

**OUTERPAD=** AUTO | *dimension* | *(pad-options)*
specifies the amount of extra space that is added outside the legend border.

- **AUTO**
  specifies that the default outside padding for this component is used.

- **dimension**
  specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend border.

- **(pad-options)**
  a space-separated list of one or more name-value pair pad options, enclosed in parentheses.
  The list can include one or more of the following pad options:

  - **LEFT=** *dimension*
    specifies the amount of extra space added to the left side.

  - **RIGHT=** *dimension*
    specifies the amount of extra space added to the right side.

  - **TOP=** *dimension*
    specifies the amount of extra space added to the top.

  - **BOTTOM=** *dimension*
    specifies the amount of extra space added to the bottom.

- **Note**
  Sides that are not assigned padding are padded with the default amount.

- **Tip**
  Use pad-options to create non-uniform padding.

**Default** AUTO

**Note**
The default units for *dimension* are pixels.

**See** "*dimension*" on page 1480

**PAD=** *dimension* | *(pad-options)*
specifies the amount of extra space that is reserved inside the legend perimeter.

- **dimension**
  specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend perimeter.

- **(pad-options)**
  a space-separated list of one or more name-value pair pad options, enclosed in parentheses.
  The list can include one or more of the following pad options:

  - **LEFT=** *dimension*
    specifies the amount of extra space added to the left side.

  - **RIGHT=** *dimension*
    specifies the amount of extra space added to the right side.
TOP=\textit{dimension} 
\begin{itemize}
\item specifies the amount of extra space added to the top.
\end{itemize}

BOTTOM=\textit{dimension} 
\begin{itemize}
\item specifies the amount of extra space added to the bottom.
\end{itemize}

\textbf{Note} \hspace{1em} Sides that are not assigned padding are padded with the default amount.

\textbf{Tip} \hspace{1em} Use \textit{pad-options} to create non-uniform padding.

\textbf{Default} \hspace{1em} Padding for all sides is 0.

\textbf{Note} \hspace{1em} The default units for \textit{dimension} are pixels.

\textbf{See} \hspace{1em} “\textit{dimension}” on page 1480

\textbf{TITLE=}“\textit{string}” 
\begin{itemize}
\item specifies the title of the legend.
\end{itemize}

\textbf{Default} \hspace{1em} No title

\textbf{Requirement} \hspace{1em} \textit{string} must be enclosed in quotation marks.

\textbf{Interaction} \hspace{1em} When the title is placed to the left of the legend, if TITLEBORDER=TRUE is in effect, no separator is displayed between the title and the legend. If TITLEBORDER=FALSE is in effect in that case, a separator is displayed.

\textbf{TITLEATTRS=}\textit{style-element} | \textit{style-element (text-options)} | (\textit{text-options}) 
\begin{itemize}
\item specifies the color and font attributes of the legend title.
\end{itemize}

\textbf{Default} \hspace{1em} The GraphLabelText style element.

\textbf{Interactions} \hspace{1em} For this option to have any effect, the \textbf{TITLE=} option must also be specified.

\textbf{Interaction} \hspace{1em} If one or more text options are specified and they do not include all the font properties (color, family, size, weight, style), then non-specified properties will be derived from the GraphLabelText style element.

\textbf{Note} \hspace{1em} When you specify \textit{style-element}, only the style attributes COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT are used.

\textbf{See} \hspace{1em} “General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}.

\textbf{“Text Options” on page 1525} for available \textit{text-options}.

\textbf{VALIGN=}\textit{CENTER} | \textit{TOP} | \textit{BOTTOM} | \textit{number} 
\begin{itemize}
\item specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.
\end{itemize}

\textbf{number} \hspace{1em} specifies an explicit position in the containing layout.
The number specification can range from 0 to 1. The number represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

The legend is effectively anchored at its center. Zero corresponds to the containing layout's bottom edge plus half the legend height. Similarly, one corresponds to the containing layout's top edge minus half the legend height.

This option is available only when this statement is nested within an overlay-type or REGION layout. It is ignored if the parent layout is not an overlay-type or REGION layout.

If LOCATION= OUTSIDE, then the VALIGN= and HALIGN= options cannot both be set to CENTER.

This option is ignored when LOCATION=INSIDE, the AUTOALIGN= option is not NONE, and the parent layout is an overlay-type layout.

VALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the legend values.

When you specify style-element, only the style attributes COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT are used.

To implement an axis legend, all of the following requirements must be met:

- An AXISLEGEND statement is used to consolidate the axis display so that a large number of tick values can be clearly displayed without collision. In an axis legend, the legend symbol is a positive integer that is generated to represent a specific axis tick position, and the legend value displays the axis tick value. In the "Example: AXISLEGEND Statement" on page 1221, the first legend symbol is the integer 1 and the first legend value is Neon SE 4dr.

- To implement an axis legend, all of the following requirements must be met:
The axis must be a discrete axis. The axis can be discrete by default, or explicitly set to discrete by setting TYPE=DISCRETE among the parent layout's axis-option settings.

The parent layout's axis options must specify NAME= to assign a name to the axis.

The parent layout's axis options must use the DISCRETEOPTS= option to specify either TICKVALUEFITPOLICY=EXTRACT or TICKVALUEFITPOLICY=EXTRACTALWAYS. EXTRACT implements an axis legend if the system estimates that a collision might occur. EXTRACTALWAYS implements an axis legend regardless of whether a collision occurs.

The AXISLEGEND statement must reference the axis name that was specified in the parent layout's NAME= axis option.

When all of these requirements are satisfied, the tick values of the named discrete axis can be replaced with consecutive positive integers. The axis legend displays those integer values and the corresponding tick values that the integers represent.

Within an overlay-type layout, when an axis legend is placed inside the plot area with LOCATION=INSIDE,

- The axis legend is always placed on top of plot lines and markers.
- By default, its background is fully transparent (OPAQUE=FALSE), meaning that underlying lines, markers, and data labels show through the legend.
- Its position can be controlled with the AUTOALIGN= option, or with the HALIGN= and VALIGN= options.

Within an overlay-type layout, when an axis legend is placed outside the plot area with LOCATION=OUTSIDE,

- By default, its background is fully opaque (OPAQUE=TRUE).
- Its position can be controlled with the HALIGN= and VALIGN= options.

When an axis legend is placed within nested layouts, it might be necessary to do one of the following to obtain the desired legend organization:

- use the ACROSS= option and also set ORDER=ROWMAJOR
- use the DOWN= option and also set ORDER=COLUMNMAJOR

Example: AXISLEGEND Statement

The following graph was generated by the “Example Program” on page 1222. The LAYOUT OVERLAY statement's XAXISOPTS= option uses NAME= to assign a name to the X axis so that it can be referenced in an AXISLEGEND statement. The LAYOUT OVERLAY statement's DISCRETEOPTS= option specifies TICKVALUEFITPOLICY=EXTRACT, which implements the axis legend if a collision occurs in the tick-value display for the X axis. The AXISLEGEND statement references the name that was assigned to the X axis.
Example Program

```sas
proc template;
define statgraph axislegend;
begingroup;
  entrytitle "Mileage for Vehicles Made by Dodge";
  layout overlay / xaxisopts=(name="xaxis"
                             discreteopts=(tickvaluefitpolicy=extract));
  barchart category=model response=mpg_highway / stat=mean;
  axislegend "xaxis";
  endlayout;
endgroup;
end;

proc sort data=sashelp.cars out=dodge; by descending mpg_highway;
  where make="Dodge";
run;

proc sgrender data=dodge template=axislegend;
run;
```

CONTINUOUSLEGEND Statement

Creates a legend with a color ramp corresponding to a range of values.

**Syntax**

```sas
CONTINUOUSLEGEND "graph-name" <options>;
```
Summary of Optional Arguments

Appearance options

BACKGROUND COLOR = \textit{style-reference | color}

specifies the color of the legend background.

\textbf{BORDER = TRUE | FALSE}

specifies whether a border is displayed around the legend.

\textbf{BORDERATTRS = style-element | style-element (line-options) | (line-options)}

specifies the attributes of the border line around the legend.

\textbf{INTEGER = TRUE | FALSE}

specifies whether only integer tick values are used in the continuous legend.

\textbf{OPAQUE = TRUE | FALSE}

specifies whether the legend background is opaque (TRUE) or transparent (FALSE).

\textbf{ORIENT = VERTICAL | HORIZONTAL}

specifies the orientation of the legend.

\textbf{OUTERPAD = AUTO | dimension | (pad-options)}

specifies the amount of extra space that is added outside the legend border.

\textbf{PAD = dimension | (pad-options)}

specifies the amount of extra space that is reserved inside the legend perimeter.

\textbf{VALUEATTRS = style-element | style-element (text-options) | (text-options)}

specifies the color and font attributes of the legend values.

Data range options

\textbf{VALUECOUNTHINT = positive-integer}

recommends a number of values for the continuous legend to use to label the data range.

Location options

\textbf{AUTOALIGN = NONE | AUTO | (location-list)}

specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.

\textbf{HALIGN = CENTER | LEFT | RIGHT | number}

specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

\textbf{LOCATION = OUTSIDE | INSIDE}

specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or REGION layout.

\textbf{VALIGN = CENTER | TOP | BOTTOM | number}

specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

Scale options

\textbf{EXTRACTSCALE = TRUE | FALSE}

specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width.

\textbf{EXTRACTSCALETYPE = DEFAULT | SCIENTIFIC}

specifies whether to extract a named scale or a scientific-notation scale.
Text options

**TITLE=** "string"

specifies the title of the legend.

**TITLEATTRS=** style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the legend title.

---

**Required Argument**

"**graph-name**"

specifies the plot to be represented by the legend. The plot is identified by the name that is assigned to it on the plot statement’s NAME= option.

- **Restriction**
  Unlike the DISCRETELEGEND statement, only a single **graph-name** can be specified.

- **Requirement**
  **graph-name** must be enclosed in quotation marks.

---

**Optional Arguments**

**AUTOALIGN=** NONE | AUTO | (location-list)

specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.

- **NONE**
  does not automatically align the legend within its parent layout.
  The legend’s position is set by the **HALIGN=** and **VALIGN=** options.
  
- **AUTO**
  within the parent layout, attempts to center the legend in the area that is farthest from any surrounding data point markers.

- **(location-list)**
  within the parent layout, restricts the legend’s possible locations to those locations in the specified **location-list**, and uses the **location-list** position that least collides with the parent layout’s other graphics features.
  The **location-list** is space-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

- **Default**
  NONE

- **Interactions**
  This option has no effect unless **LOCATION=** INSIDE.
  When **LOCATION=** INSIDE and **AUTOALIGN=** is not NONE, the **HALIGN=** and **VALIGN=** options are ignored.

- **See**
  LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type layout.

**BACKGROUNDCOLOR=** style-reference | color

specifies the color of the legend background.

- **style-reference**
  specifies a style reference in the form **style-element**:**style-attribute**.
  Only the style-attribute named COLOR or CONTRASTCOLOR is used.
**CONTINUOUSLEGEND Statement**

**Default**
The GraphLegendBackground:Color style reference.

**Interaction**
OPAQUE= TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

**BORDER=TRUE | FALSE**
specifies whether a border is displayed around the legend.

**Defaults**
FALSE in SAS 9.4M1 and earlier releases.

The GraphLegendBackground:FrameBorder style reference starting with SAS 9.4M2. If attribute FrameBorder is not defined in the active style, then FALSE is the default value.

**Tip**
The BORDERATTRS= option controls the appearance of the legend border.

**See**
“boolean ” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=style-element | style-element (line-options) | (line-options)**
specifies the attributes of the border line around the legend.

**Default**
The GraphBorderLines style element.

**Interaction**
BORDER= TRUE must be set for this option to have any effect.

**Tip**
The color of the frame around the color ramp and the color ramp tick lines is controlled by the GraphAxisLines:contrastColor style attribute.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element

“Line Options” on page 1522 for available line-options.

**EXTRACTSCALE=TRUE | FALSE**
specifies whether to extract a scale factor from the tick values and use it to reduce the tick value width. The scale can be a named scale or a scientific-notation scale. The EXTRACTSCALETYPE= option specifies the scale type. The scale used is appended to the legend title as shown in the following example.

*Total Sales (millions)*

For long legend titles, if the scale does not fit the available space, then the title is truncated, and the scale is appended to the truncated title. Ellipses indicate that the label was truncated as shown in the following example.

*Total Sales for the Fourth Quarter Of ... (millions)*

In extreme cases where the title does not fit even with truncation, the title is dropped.

**Restriction**
The scale that is extracted by the EXTRACTSCALE= option is derived from the English locale.

**Interactions**
The scale type is determined by the EXTRACTSCALETYPE= option.

If the axis label is not displayed, then the EXTRACTSCALE=TRUE option is ignored.
See "boolean" on page 1479 for other Boolean values that you can use.

**EXTRACTSCALETYPE=DEFAULT | SCIENTIFIC**

specifies whether to extract a named scale or a scientific-notation scale.

**DEFAULT**

extracts a named scale.

A named scale can be millions, billions, or trillions for values of 999 trillion or less, or a multiple of 10 (denoted as $10^n$) for values over 999 trillion. For small fractional tick values, the scale factor is set to ensure that the absolute value of the smallest value is greater than 1. The scale can be millionth, billionth, or trillionth for values of 1 trillionth or more, or a multiple of 1/10 ($10^{-n}$) for values less than 1 trillionth.

**SCIENTIFIC**

extracts a scientific-notation scale.

A scientific-notation scale is a multiple of 10 expressed as $10^n$ for values greater than 1, or a multiple of 1/10 expressed as $10^{-n}$ for values less than 1.

**HALIGN=CENTER | LEFT | RIGHT | number**

specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

**number**

specifies an explicit position in the containing layout.

**Range**

The number specification can range from 0 to 1. The number represents a fraction of the parent container’s width, where 0 is all the way to the left and 1 is all the way to the right.

**Interaction**

For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

**Tip**

The legend is effectively anchored at its center. HALIGN=0 corresponds to the containing layout's left edge plus half the legend width. Similarly, HALIGN=1 corresponds to the containing layout's right edge minus half the legend width.

**Defaults**

If LOCATION=OUTSIDE, then the default is RIGHT.

If LOCATION=INSIDE, then the default is CENTER.

**Restriction**

This option is available only when this statement is nested within an overlay-type or REGION layout.

**Interactions**

If LOCATION=OUTSIDE, then the HALIGN= and VALIGN= options cannot both be set to CENTER.

This option is ignored when LOCATION=INSIDE, the AUTOALIGN= option is not NONE, and the parent layout is an overlay-type layout.

**See**

LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

**INTEGER=TRUE | FALSE**

specifies whether only integer tick values are used in the continuous legend.
CONTINUOUSLEGEND Statement

Note: This feature applies to SAS 9.4M3 and to later releases.

Default FALSE

Restriction This option applies only to smooth color gradients. For leveled gradients such as those used with contour plots of type FILL, LINEFILL or LABELEDLINEFILL, this option is ignored. See CONTOURTYPE=.

Interaction INTEGER=TRUE is ignored when there are no integer values between the minimum and maximum legend values.

LOCATION=OUTSIDE | INSIDE
specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or REGION layout.

Default OUTSIDE

Restriction This option has effect only when the legend statement appears within a 2-D overlay-type layout and there is at least one standalone plot statement with XY axes that is referenced by the legend.

Interactions The actual position is determined by the settings for the LOCATION=, AUTOALIGN=, HALIGN=, and VALIGN= options.

Within an overlay-type layout, if the ORIENT= option is not set, then the orientation changes depending on the actual position. If LOCATION=OUTSIDE and the legend is right or left of the plot, then the orientation is vertical. If LOCATION=OUTSIDE and the legend is above or below the plot, then the orientation is horizontal.

See LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

OPAQUE=TRUE | FALSE
specifies whether the legend background is opaque (TRUE) or transparent (FALSE).

Default FALSE

Interaction When this option is set to FALSE, the background color is not used.

See “boolean” on page 1479 for other Boolean values that you can use.

ORIENT=VERTICAL | HORIZONTAL
specifies the orientation of the legend.

Default VERTICAL

Restriction In order to use ORIENT=HORIZONTAL when the continuous legend is in an OVERLAY or REGION layout, you must place the legend inside the graph area (LOCATION=INSIDE).
Tip To orient the legend horizontally outside of the graph area in an OVERLAY or REGION layout, use LOCATION=OUTSIDE and VALIGN=BOTTOM instead.

OUTERPAD=AUTO | dimension | (pad-options)
specifies the amount of extra space that is added outside the legend border.

AUTO
specifies that the default outside padding for this component is used.

dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend border.

(pad-options)
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.
The list can include one or more of the following pad options:

LEFT=dimension
specifies the amount of extra space added to the left side.

RIGHT=dimension
specifies the amount of extra space added to the right side.

TOP=dimension
specifies the amount of extra space added to the top.

BOTTOM=dimension
specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.

Tip Use pad-options to create non-uniform padding.

Default AUTO

Note The default units for dimension are pixels.

See “dimension” on page 1480

PAD=dimension | (pad-options)
specifies the amount of extra space that is reserved inside the legend perimeter.

dimension
specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend perimeter.

(pad-options)
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.
The list can include one or more of the following pad options:

LEFT=dimension
specifies the amount of extra space added to the left side.

RIGHT=dimension
specifies the amount of extra space added to the right side.
TOP=\textit{dimension}  
specifies the amount of extra space added to the top.

\textbf{BOTTOM=\textit{dimension}}  
specifies the amount of extra space added to the bottom.

\textbf{Note}  
Sides that are not assigned padding are padded with the default amount.

\textbf{Tip}  
Use \textit{pad-options} to create non-uniform padding.

\textbf{Default}  
Padding for all sides is 0.

\textbf{Note}  
The default units for \textit{dimension} are pixels.

\textbf{See}  
“\textit{dimension}” on page 1480

\textbf{TITLE=}“\textit{string}”  
specifies the title of the legend. The title is placed below the legend body.

\textbf{Default}  
No title

\textbf{Requirement}  
\textit{string} must be enclosed in quotation marks.

\textbf{Interaction}  
When the title is placed to the left of the legend, if \textbf{TITLEBORDER=}TRUE is in effect, no separator is displayed between the title and the legend. If \textbf{TITLEBORDER=}FALSE is in effect in that case, a separator is displayed.

\textbf{TITLEATTRS=}\textit{style-element} | \textit{style-element (text-options)} | \textit{(text-options)}  
specifies the color and font attributes of the legend title.

\textbf{Default}  
The GraphLabelText style element.

\textbf{Interactions}  
For this option to have any effect, the \textbf{TITLE=} option must also be specified.

If one or more text options are specified and they do not include all the font properties (color, family, size, weight, style), then non-specified properties will be derived from the GraphLabelText style element.

\textbf{Note}  
When you specify \textit{style-element}, only the style attributes \textit{COLOR}, \textit{FONTFAMILY}, \textit{FONTSIZE}, \textit{FONTSTYLE}, and \textit{FONTWEIGHT} are used.

\textbf{See}  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}.

“Text Options” on page 1525 for available \textit{text-options}.

\textbf{VALIGN=}\textbf{CENTER} | \textbf{TOP} | \textbf{BOTTOM} | \textit{number}  
specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

\textit{number}  
specifies an explicit position in the containing layout.
Range

The number specification can range from 0 to 1. The number represents a fraction of the parent container's height, where 0 is on the bottom and 1 is on the top.

Interaction

For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

Tip

The legend is effectively anchored at its center. Zero corresponds to the containing layout's bottom edge plus half the legend height. Similarly, one corresponds to the containing layout's top edge minus half the legend height.

Default

CENTER

Restriction

This option is available only when this statement is nested within an overlay-type or REGION layout. It is ignored if the parent layout is not an overlay-type or REGION layout.

Interactions

If LOCATION=OUTSIDE, then the VALIGN= and HALIGN= options cannot both be set to CENTER.

This option is ignored when LOCATION=INSIDE, the AUTOALIGN= option is not NONE, and the parent layout is an overlay-type layout.

See

LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

VALUEATTRS=style-element | style-element (text-options) | (text-options)

specifies the color and font attributes of the legend values.

Default

The GraphValueText style element.

Interaction

If one or more text options are specified and they do not include all the font properties (color, family, size, weight, style), then non-specified properties will be derived from the GraphLabelText style element.

Note

When you specify style-element, only the style attributes COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT are used.

See

“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

VALUECOUNTHINT=positive-integer

recommends a number of values for the continuous legend to use to label the data range.

Default

6

Restriction

The associated plot must be displayed with smooth gradient for this option to have any effect. For example, in a contour plot, CONTOURTYPE must be set to GRADIENT or LINEGRADIENT.
Requirement  *positive-integer* must be greater than zero.

Note  The legend uses even intervals to label the range.

**Details**

A continuous legend consists of a color ramp and a numeric scale indicating color values.

In the following figure, the continuous legend references a contour plot with a fixed number of levels. The color ramp and legend values automatically reflect these discrete levels. The legend option `VALUECOUNTINT=` has no effect.

In this next figure, the continuous legend references a contour plot with a continuous gradient. The number of legend values displayed is automatically determined by the legend, or decided by the contributing plot, such as the `CONTOURPLOTPARM` with `CONTOUTYPE=FILL`. For plots with a continuous gradient, you can use the legend option `VALUECOUNTINT=` to control how many legend values appear. (This option is ignored if there is no gradient.)
Within an overlay-type layout, when a continuous legend is placed inside the plot area with \texttt{LOCATION=INSIDE},

- It is always placed on top of plot lines and markers.
- By default, its background is fully transparent (\texttt{OPAQUE=FALSE}), meaning that underlying lines, markers, and data labels show through the legend.
- Its position can be controlled with the \texttt{AUTOALIGN=} option, or with the \texttt{HALIGN=} and \texttt{VALIGN=} options.

Within an overlay-type layout, when a continuous legend is placed outside the plot area with \texttt{LOCATION=OUTSIDE},

- By default, its background is fully opaque (\texttt{OPAQUE=TRUE}).
- Its position can be controlled with the \texttt{HALIGN=} and \texttt{VALIGN=} options.

Example: CONTINUOUSLEGEND Statement

The following graph was generated by the “Example Program” on page 1233:
Example Program

```sas
proc template;
define statgraph continuouslegend;
begingraph;
  entrytitle "Height and Weight Distribution";
  layout overlay;
  scatterplot x=height y=weight /
    markercolorgradient=density
    markerattrs=(symbol=squarefilled size=6px)
    name="scatter";
  continuouslegend "scatter" / orient=vertical
    location=outside valign=center halign=right
    valuecounthint=10 title="Density";
  endlayout;
endgraph;
end;
run;
ods graphics / reset width=475px;
proc sgrender data=sashelp.gridded(where=(count>0))
  template=continuouslegend;
run;
```
DISCRETELEGEND Statement

MERGEDLEGEND Statement

Creates a legend with entries that refer to plots, or group values, or both legend items.

Restrictions:
The MERGEDLEGEND statement can be used for grouped plots only.
The MERGEDLEGEND statement supports only line and marker overlays.

Notes:
Often the data in a plot is classified by a group variable. Or, multiple columns of data are plotted in the same graph. These groups or columns are represented in the graph by different color or line patterns or marker symbols. In these cases, a DISCRETELEGEND can be added to the graph to help decode the data. The MERGEDLEGEND statement can be used to consolidate legend entries when the graph displays grouped data for two plots. The MERGEDLEGEND statement must specify exactly two names that reference the source for the legend entry values.

Prior to SAS 9.4M3, when a discrete attribute map is used for group values in a plot that contributes to a discrete legend and attributes are overridden in the plot statement, the attributes of some plot features and their corresponding legend items might not match. Starting with SAS 9.4M3, the attributes of the legend items always match the attributes of the corresponding plot features.

See: "LEGENDITEM Statement" on page 1252

Syntax

DISCRETELEGEND "graph-name-1" | "legend-item-name-1" | "discrete-attr-name-1"
<"graph-name-2" | "legend-item-name-2" | "discrete-attr-name-2"…> </options>;

MERGEDLEGEND "graph-name" | "legend-item-name" | "discrete-attr-name"
"graph-name" | "legend-item-name" | "discrete-attr-name"
</options>;

Summary of Optional Arguments

Appearance options

ACROSS=positive-integer
specifies the number of legend entries that are placed horizontally before the next row begins.

AUTOITEMSIZE=TRUE | FALSE
specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size used for the legend entry labels.

BACKGROUNDCOLOR=style-reference | color
specifies the color of the legend background.

BORDER=TRUE | FALSE
specifies whether a border is displayed around the legend.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the border line around the legend.

**DISPLAYCLIPPED**=

**DOWN**=

**FILLITEMOUTLINE**=

**ITEMSIZE**=(

**OPAQUE**=

**ORDER**=

**OUTERPAD**=

**PAD**=

**SORTBY**=

**SORTORDER**=

**TITLEBORDER**=

**VALUEATTRS**=

**ADDITIONALNAMES**=

**EXCLUDE**=(

**TYPE**=

**AUTOALIGN**=

**HALIGN**=

**Location options**

**AUTOALIGN**=

**HALIGN**=
specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

LOCATION=OUTSIDE | INSIDE
specifies whether the legend appears inside or outside the plot area when the legend is specified within an overlay-type or REGION layout.

VALIGN=CENTER | TOP | BOTTOM | number
specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

Text options

TITLE="string"
specifies the title of the legend.

TITLEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the legend title.

Required Arguments

"graph-name"
references one or more unique names that are specified by the NAME= option of a plot statement. These names control which plots contribute to the legend, and the order in which the legend entries occur. If a contributing plot uses a GROUP= option, then there is a legend entry for each group value that is present in the data.

Requirements Each graph-name must be enclosed in quotation marks.

Each plot that is referenced must be able to be identified by the legend entries. For example, a filled CONTOURPLOTPARM plot cannot be specified in the DISCRETELEGEND or MERGEDLEGEND statement because it requires a continuous legend.

"legend-item-name"
references one or more unique values specified by the NAME= option of a LEGENDITEM statement. Each legend-item-name must be enclosed in quotation marks.

See "LEGENDITEM Statement" on page 1252

"discrete-attr-name"
references one or more unique values that are specified by the NAME= option in a DISCRETEATTRMAP statement. The discrete attribute map that the name references contributes all of its value statements as legend items, regardless of whether they match the data.

See "DISCRETEATTRMAP Statement" on page 1425

Optional Arguments

ACROSS=positive-integer
specifies the number of legend entries that are placed horizontally before the next row begins. A legend entry typically consists of two parts, such as a marker symbol and an associated value.
The entries are placed to best fit the available area. This “best fit” approach works only when the legend is nested in the template’s outermost layout.

This option is ignored if ORDER= COLUMNMAJOR

This option is ignored when the DISCRETELEGEND statement is in a LAYOUT GLOBALLEGEND block and the TYPE=ROW option is in effect for the layout.

ADDITIONALNAMES=("graph-name" | "legend-item-name" | "discrete-attr-name" <"graph-name" | "legend-item-name" | "discrete-attr-name"> ...) specifies additional legend items that are to be added to the two items that are required in the MERGEDLEGEND statement.

Note: This option is valid starting in SAS 9.4M1.

The items from the sources that are specified in this option are not included in the merging of the legend items. They are appended to the legend after the items from the sources that are specified in the required arguments are merged.

This option applies to the MERGEDLEGEND statement only.

The list of names must be enclosed in parentheses.

Each name must be enclosed in quotation marks and separated from adjacent values by a blank space.

AUTOALIGN=NONE | AUTO | (location-list) specifies whether the legend is automatically aligned within its parent layout when nested within an overlay-type layout.

NONE
does not automatically align the legend within its parent layout.

The legend’s position is set by the HALIGN= and VALIGN= options.

AUTO
within the parent layout, attempts to center the legend in the area that is farthest from any surrounding data point markers.

(location-list)
within the parent layout, restricts the legend’s possible locations to those locations in the specified location-list, and uses the location-list position that least collides with the parent layout’s other graphics features.

The location-list is space-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Default NONE
Restriction
AUTO is available only when the parent layout contains a scatter plot. Otherwise, it is ignored.

Interactions
This option has no effect unless LOCATION= INSIDE.

When LOCATION=INSIDE and AUTOALIGN= is not NONE, the HALIGN= and VALIGN= options are ignored.

Note
Prior to SAS 9.4M5, the legend might overlap other legends or insets. Starting with SAS 9.4M5, alignments that cause the legend to overlap another legend or inset are avoided.

See LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type layout.

AUTOITEMSIZE=TRUE | FALSE
specifies that all markers, lines, and filled symbols in the legend are sized in proportion to the font size used for the legend entry labels. These proportional sizes take effect regardless of the size reported by the plot or LEGENDITEM. The line segments are drawn as deemed appropriate by the legend, regardless of the line thickness reported by the plot.

Default FALSE

Interaction When set to TRUE, this setting considers the font size in effect from the VALUEATTRS= option.

See “boolean ” on page 1479 for other Boolean values that you can use.

BACKGROUND COLOR=style-reference | color
specifies the color of the legend background.

style-reference specifies a style reference in the form style-element:style-attribute.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

Default The GraphLegendBackground:Color style reference.

Interaction OPAQUE= TRUE must be in effect for the color to be seen. By default, OPAQUE=FALSE.

BORDER=TRUE | FALSE
specifies whether a border is displayed around the legend.

Defaults TRUE in SAS 9.4M1 and earlier releases.

The GraphLegendBackground:FrameBorder style reference starting with SAS 9.4M2. If attribute FrameBorder is not defined in the active style, then TRUE is the default value.

Tip The BORDERATTRS= option controls the appearance of the legend border.

See “boolean ” on page 1479 for other Boolean values that you can use.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the border line around the legend.
The GraphBorderLines style element.

BORDER= TRUE must be set for this option to have any effect.

See "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element

"Line Options" on page 1522 for available line-options.

**DISPLAYCLIPPED=TRUE | FALSE**

specifies whether the legend is displayed when any portion of the legend cannot be rendered entirely. Based on the legend contents and placement, or when the graph size is reduced, parts of the legend (title, legend symbol, or legend value) might be clipped (truncated). When clipping occurs and this option is set to FALSE, the entire legend is removed from the graph. The space for that legend is then reclaimed by the remainder of the graph. When this option is set to TRUE, the legend always appears, even if some parts of it have been clipped.

Default FALSE

See "boolean " on page 1479 for other Boolean values that you can use.

**DOWN=positive-integer**

specifies the number of legend entries that are placed vertically before the next column begins. A legend entry typically consists of two parts, such as a marker symbol and an associated value.

Default The entries are placed to best fit the available area. This “best fit” approach works only when the legend is nested in the template’s outermost layout.

Restriction This option is ignored if ORDER= ROWMAJOR

Interaction This option is ignored when the DISCRETELEGEND statement is in a LAYOUT GLOBALLEGEND block.

**EXCLUDE=(item-names)**

specifies a list of legend entries to exclude from the display.

Default No items are excluded.

Requirement Each item name must be enclosed in quotation marks and separated from adjacent names by a space.

Note When the specified names are compared with the legend entry names, leading blanks are honored and trailing blanks are ignored.

Tip For plots with groups, you can exclude specific group values.

Example The following example excludes items Truck and Wagon from the legend.

```
exclude=("Truck" "Wagon")
```

**FILLITEMOUTLINE=AUTO | ON**

specifies whether the fill swatches are outlined only when enabled by the contributing statements or are always outlined.
Note: This feature applies to SAS 9.4M3 and to later releases.

AUTO
honors the DISPLAY= option settings for the contributing plot or the FILLDISPLAY= setting of the legend item.

ON
always outlines the fill swatches.

Default ON

Restriction This option is valid only in the DISCRETELEGEND statement.

Requirement TYPE= on page 1246 must be FILL, FILLCOLOR, or ALL for this option to have any effect.

Note The outlines are always 1 pixel wide with a solid pattern.

Tip To display color swatches with no outline when DISPLAY= includes OUTLINE, specify FILLITEMOUTLINE=AUTO and TYPE=FILLCOLOR.

HALIGN= CENTER | LEFT | RIGHT | number
specifies the horizontal alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

number specifies an explicit position in the containing layout.

Range The number specification can range from 0 to 1. The number represents a fraction of the parent container’s width, where 0 is all the way to the left and 1 is all the way to the right.

Interaction For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

Tip The legend is effectively anchored at its center. HALIGN=0 corresponds to the containing layout's left edge plus half the legend width. Similarly, HALIGN=1 corresponds to the containing layout's right edge minus half the legend width.

Default CENTER

Restriction This option is available only when this statement is nested within an overlay-type or REGION layout.

Interactions If LOCATION= OUTSIDE, then the HALIGN= and VALIGN= options cannot both be set to CENTER.

This option is ignored when LOCATION=INSIDE, the AUTOALIGN= option is not NONE, and the parent layout is an overlay-type layout.

See LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.
ITEMSIZE=(size-options)
specifies the size of specific types of items that are in a discrete or merged legend.

Note: This option is valid starting in SAS 9.4M1.

The following size options are supported:

FILLASPECTRATIO=AUTO | GOLDEN | positive-number
specifies the aspect ratio for the fill swatches.

Note: This feature applies to SAS 9.4M3 and to later releases.

AUTO
uses an equal width and height for color swatches with solid color fills, or uses the golden ratio for swatches with pattern fills.

GOLDEN
specifies the golden ratio of 1.618 (width = 1.618 * height) for both solid color and pattern fill swatches.

positive-number
specifies a custom aspect ratio.

Default AUTO
Restrictions This option is valid only in the DISCRETELEGEND statement.
This option does not apply to bubble plot fill color swatches (filled circle).
Interactions The legend must display fill entries for this option to have any effect.
This option is ignored when AUTOITEMSIZE=TRUE.

When FILLASPECTRATIO= is specified but neither FILLHEIGHT= nor HEIGHTSCALE= is specified, the height for color swatches with a solid fill and color swatches with a pattern fill differ.

Note FILLHEIGHT=, HEIGHTSCALE=, and FILLASPECTRATIO= also apply to fill-pattern color swatches with no outline.
Tip Use FILLHEIGHT= or HEIGHTSCALE= to specify the height.

FILLHEIGHT=AUTO | BIG | dimension | style-reference
specifies the height of the fill swatches.

Note: This feature applies to SAS 9.4M3 and to later releases.

AUTO
enables the SAS system to determine the fill color swatch height.
BIG
specifies sizes that were developed as defaults for SAS Visual Analytics, 
which results in color swatches that are larger than the SAS system 
defaults.

dimension
specifies a custom height for the fill swatches.
See  “dimension” on page 1480

style-reference
specifies a style attribute that controls the height of the fill swatches.
The style reference must provide a valid dimensional value.
See  “style-reference ” on page 1482

Default AUTO
Restriction This option is valid only in the DISCRETELEGEND statement.
Interactions The legend must display fill entries for this option to have any 
effect.

This option is ignored when AUTOITEMSIZE=TRUE.

Tip Use FILLASPECTRATIO= to specify the width.

HEIGHTSCALE=positive-number
specifies a scale factor that is to be applied to the fill swatch height.
Values greater than 1 increase the height of the fill swatches, and values less 
than 1 reduce the height.

Note: This feature applies to SAS 9.4M3 and to later releases.

Default 1
Restriction This option is valid only in the DISCRETELEGEND statement.
Interactions The legend must display fill entries for this option to have any 
effect.

This option is ignored when AUTOITEMSIZE=TRUE.

Tips Use FILLHEIGHT= to change the base height.
Use FILLASPECTRATIO= to specify the width.

LINELENGTH=dimension
specifies the length of the line glyph for line entries in the legend.

Default 48 pixels
Interaction The legend must display line entries for this option to have any 
effect.

See  “dimension” on page 1480
LOCATION=OUTSIDE | INSIDE
specifies whether the legend appears inside or outside the plot area when the
legend is specified within an overlay-type or REGION layout.

Default OUTSIDE

Restriction This option has effect only when the legend statement appears
within an overlay-type or REGION layout and at least one stand-
alone plot statement is referenced by the parent layout.

Interactions The actual position is determined by the settings for the
LOCATION=, AUTOALIGN=, HALIGN=, and VALIGN= options.

If this option is set to OUTSIDE, then the HALIGN= and VALIGN= options must specify a keyword (LEFT, RIGHT, or CENTER). The number setting for the alignment is invalid when the legend is
positioned outside of the plot area.

See LAYOUT OVERLAY on page 149 for more information about how
child positions are determined in an overlay-type or REGION layout.

OPAQUE=TRUE | FALSE
specifies whether the legend background is opaque (TRUE) or transparent
(FALSE).

Default FALSE

Interaction When this option is set to FALSE, the background color is not used.

See “boolean ” on page 1479 for other Boolean values that you can use.

ORDER=ROWMAJOR | COLUMNMAJOR
specifies whether legend entries are organized into rows or into columns.

Default ROWMAJOR

Interaction If ORDER=ROWMAJOR, then use the ACROSS= option to limit the
number of entries in a row. If ORDER=COLUMNMAJOR, then use
the DOWN= option to limit the number of entries in a column.

OUTERPAD=AUTO | dimension | (pad-options)
specifies the amount of extra space that is added outside the legend border.

AUTO specifies that the default outside padding for this component is used.

dimension specifies a dimension to use for the extra space at the left, right, top, and
bottom of the legend border.

(pad-options)
a space-separated list of one or more name-value pair pad options, enclosed
in parentheses.

The list can include one or more of the following pad options:

LEFT=dimension specifies the amount of extra space added to the left side.

RIGHT=dimension specifies the amount of extra space added to the right side.
TOP=\texttt{dimension}

specifies the amount of extra space added to the top.

\textbf{BOTTOM=\texttt{dimension}}

specifies the amount of extra space added to the bottom.

\begin{itemize}
  \item \textbf{Note} Sides that are not assigned padding are padded with the default amount.
  \item \textbf{Tip} Use \texttt{pad-options} to create non-uniform padding.
\end{itemize}

\textbf{Default:} AUTO

\textbf{Notes:} The default units for \texttt{dimension} are pixels.

Starting with \texttt{SAS 9.4M1}, the default padding between the discrete legend and the plot area (including the axes) is increased to 10 pixels, depending on the context. If the new default padding is not desirable, then use the \texttt{OUTERPAD=} option to adjust it.

\textbf{See:} “\texttt{dimension}” on page 1480

\textbf{PAD=} \texttt{dimension} | (\texttt{pad-options})

specifies the amount of extra space that is reserved inside the legend perimeter.

\begin{itemize}
  \item \texttt{dimension} specifies a dimension to use for the extra space at the left, right, top, and bottom of the legend perimeter.
  \item \texttt{(pad-options)} a space-separated list of one or more name-value pair pad options, enclosed in parentheses.
\end{itemize}

The list can include one or more of the following pad options:

\begin{itemize}
  \item \texttt{LEFT=} \texttt{dimension} specifies the amount of extra space added to the left side.
  \item \texttt{RIGHT=} \texttt{dimension} specifies the amount of extra space added to the right side.
  \item \texttt{TOP=} \texttt{dimension} specifies the amount of extra space added to the top.
  \item \texttt{BOTTOM=} \texttt{dimension} specifies the amount of extra space added to the bottom.
\end{itemize}

\begin{itemize}
  \item \textbf{Note} Sides that are not assigned padding are padded with the default amount.
  \item \textbf{Tip} Use \texttt{pad-options} to create non-uniform padding.
\end{itemize}

\textbf{Default:} Padding for all sides is 0.

\textbf{Note:} The default units for \texttt{dimension} are pixels.

\textbf{See:} “\texttt{dimension}” on page 1480

\textbf{SORTBY=} \texttt{LABEL} | \texttt{TEXT}

specifies whether text legend items are sorted by label or by text.
Note: This feature applies to SAS 9.4M3 and to later releases.

Default: LABEL

Restriction: This option is valid only in the DISCRETELEGEND statement.

Interaction: This option has an effect only when sorting is performed, TYPE=TEXT is specified in this DISCRETELEGEND statement, and the legend items are defined in either a LEGENDTEXTITEM statement or in a LEGENDITEM statement that specifies TYPE=TEXT.

SORTORDER=AUTO | REVERSEAUTO | ASCENDINGFORMATTED | DESCENDINGFORMATTED

specifies the sort order to use for the legend entry labels.

AUTO

presents groups of legend entries in the order in which they are listed in the legend statement.

The internal ordering of the entries is derived from the constituent plot-statement options.

REVERSEAUTO

presents groups of legend entries in the reverse order in which they are listed in the legend statement.

Note: This feature applies to SAS 9.4M5 and to later releases.

Tip: This option is useful for grouped, stacked vertical bar charts when you want the legend entries to be drawn in the same order as the bar group segments.

ASCENDINGFORMATTED

DESCENDINGFORMATTED

performs a linguistic sort on the group items. It has the same effect as sorting the input data, but the data is not changed.

Entries from separate plots, and separate legend items are combined and ordered as a single list.

Default: AUTO. Groups of legend entries are presented in the order in which they are listed in the legend statement. The internal ordering of the entries is derived from the constituent plot-statement options.

Interaction: This option overrides the order that is set by any constituent plot statement’s GROUPORDER= option.

TITLE="string"

specifies the title of the legend. The title is placed to the left of the legend body, except in the following cases:

- The legend contains two or more rows of items.
- The legend is in a nested layout.
- The legend is in an OVERLAYEQUATED layout.
The legend title length exceeds the space that is available on the left side of the legend. In those cases, the title is placed above the legend body.

Default  
No title

Requirement  
*string* must be enclosed in quotation marks.

Interaction  
When the title is placed to the left of the legend, if TITLEBORDER=TRUE is in effect, no separator is displayed between the title and the legend. If TITLEBORDER=FALSE is in effect in that case, a separator is displayed.

**TITLEATTRS=** 
*style-element | style-element (text-options) | (text-options)*

specifies the color and font attributes of the legend title.

Default  
The GraphLabelText style element.

Interactions  
For this option to have any effect, the **TITLE=** option must also be specified.

If one or more text options are specified and they do not include all the font properties (color, family, size, weight, style), then non-specified properties will be derived from the GraphLabelText style element.

Note  
When you specify *style-element*, only the style attributes COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT are used.

See  
“General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element*.

“Text Options” on page 1525 for available *text-options*.

**TITLEBORDER=** 
TRUE | FALSE

specifies a border around the legend title that separates it from the legend body.

Default  
FALSE

Interactions  
Prior to **SAS 9.4M3**, when LOCATION=INSIDE, BORDER=TRUE and the TITLE= option must also be in effect for this option to have any effect. When LOCATION=OUTSIDE, the TITLE= option must also be in effect. The BORDER= option has no effect on the title border in that case.

Starting with **SAS 9.4M3**, BORDER=TRUE and the TITLE= option must also be in effect for this option to have any effect. The LOCATION= option has no effect on the title border in that case.

Tip  
The line attributes of the title border are set by the BORDERATTRS= options.

See  
“boolean” on page 1479 for other Boolean values that you can use.

**TYPE=** 
ALL | FILL | FILLCOLOR | LINE | LINECOLOR | LINEPATTERN | MARKER | MARKERCOLOR | MARKERSYMBOL | TEXT

specifies which visual attributes to display for legend entries in the legend.
Note: TEXT is valid starting with SAS 9.4M3.

The TYPE= option can be used as a filter. If a statement contributing to the legend does not have any visual attributes that match the TYPE specified, then the legend does not display any entries from that statement.

Some keywords can be used to create specialized legends that display a single visual attribute. For example, keywords FILLCOLOR or MARKERSYMBOL result in the display of a single attribute. Other keywords (for example, FILL, LINE, or MARKER) result in legends that display a set of visual attributes. For example, keyword LINE results in the display of both line color and line pattern for legend entries that include lines in their display.

If this option is set to LINEPATTERN or MARKERSYMBOL, then a filled symbol is drawn using the same text color as the color used for the legend entry labels. The symbol is sized automatically, as if the AUTOITEMSIZE= option is set to TRUE. For keywords FILLCOLOR, LINECOLOR, and MARKERCOLOR, the filled symbols are drawn as outlined color swatches. The outline is 1 pixel wide, and its color is controlled by the CONTRASTCOLOR attribute of the GraphOutlines style element.

**Default**

ALL

**Restrictions**

This option is valid only in the DISCRETELEGEND statement.

A LEGENDITEM statement can be referenced only from a discrete legend of the same attribute type or of an overlapping attribute type. For legends that display multiple visual attributes (use both colors and marker symbols, for example), the default visual properties are derived from the GraphDataDefault style-element.

Starting with SAS 9.4M3, the LEGENDTEXTITEMS statement can be referenced only from a discrete legend of type TEXT or ALL.

**VALIGN=**CENTER | TOP | BOTTOM | number

specifies the vertical alignment of the legend within its parent layout when nested within an overlay-type or REGION layout.

**number**

specifies an explicit position in the containing layout.

**Range**

The number specification can range from 0 to 1. The number represents a fraction of the parent container’s height, where 0 is on the bottom and 1 is on the top.

**Interaction**

For a number setting to take effect, LOCATION=INSIDE must be set. A number setting is invalid on this option when LOCATION=OUTSIDE.

**Tip**

The legend is effectively anchored at its center. Zero corresponds to the containing layout's bottom edge plus half the legend height. Similarly, one corresponds to the containing layout's top edge minus half the legend height.

**Default**

CENTER
Restriction
This option is available only when this statement is nested within an overlay-type or REGION layout. It is ignored if the parent layout is not an overlay-type or REGION layout.

Interactions
If LOCATION= OUTSIDE, then the VALIGN= and HALIGN= options cannot both be set to CENTER.

This option is ignored when LOCATION=INSIDE, the AUTOALIGN= option is not NONE, and the parent layout is an overlay-type layout.

See LAYOUT OVERLAY on page 149 for more information about how child positions are determined in an overlay-type or REGION layout.

VALUEATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font attributes of the legend values.

Default
The GraphValueText style element.

Interaction
If one or more text options are specified and they do not include all the font properties (color, family, size, weight, style), then non-specified properties will be derived from the GraphLabelText style element.

Note
When you specify style-element, only the style attributes COLOR, FONTFAMILY, FONTSIZE, FONTSTYLE, and FONTWEIGHT are used.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

Details
A discrete legend consists of one or more units called legend entries. Each legend entry consists of a legend symbol and a legend value. The legend symbol is typically a marker, line, or filled symbol that represents a specific area in the plot. The legend value is descriptive text that is derived from group values, or that is assigned by the plot’s LEGENDLABEL= option or by the LEGENDITEM statement’s LABEL= option. To specify a discrete legend, you can use the DISCRETELEGEND statement. To consolidate legend entries for common grouped values that are represented by two separate plots, you can use the MERGEDLEGEND statement.

For grouped plots, a discrete legend represents all of the values that are present in the data. To ensure that legend entries are displayed for group values, regardless of whether those values are present in the data, you can use a discrete attribute map as described in “Displaying Legend Entries for Group Values That Are Not in the Data” on page 1431.

In the DISCRETELEGEND statement, you can specify one or more names that reference the source for the legend entry values. You can use the TYPE= option to control the visual attributes for the legend display.

For legend items that represent fills or colors without a specific shape, a filled symbol with a one-pixel-wide, solid outline is used to represent the data values. If the feature being represented by the legend item has an outline, then the default outline color is derived from the feature’s outline color. If the feature being represented is an outline only or is a filled outlined marker, then the outline...
thickness is derived from the plot. If the feature does not have an outline, then the
default outline color for its corresponding legend entry is derived from the
GraphOutline style element.

You can use the MERGEDLEGEND statement to consolidate lines and marker
symbols from discrete legend entries when a graph displays grouped values for
exactly two plots. With a MERGEDLEGEND statement, the legend values from the
group variables in two plots are compared. For each common value, the
respective legend lines and marker symbols are combined, and only one legend
entry is created for each matching set of group values. You can use the
MERGEDLEGEND statement only for grouped plots. You must specify two names
that reference the source for the legend entry values.

When a discrete legend is placed inside the plot area of an overlay type layout and
LOCATION= INSIDE, the following conditions apply:
- The discrete legend is always placed on top of plot lines and markers.
- By default, the background of the discrete legend is fully transparent
  (OPAQUE= FALSE). Underlying lines, markers, and data labels show through
  the legend.
- You can control the position of the discrete legend with the AUTOALIGN= option,
or with the HALIGN= and VALIGN= options. (The AUTOALIGN= option is not
  available in a LAYOUT OVERLAY3D statement.)

When a discrete legend is placed outside the plot area of an overlay type layout and
LOCATION=OUTSIDE, the following conditions apply:
- By default, the background of the discrete legend is fully opaque
  (OPAQUE=TRUE).
- You can control the position of the discrete legend with the HALIGN= and
  VALIGN= options.

When a discrete legend is placed within nested layouts, you might need to do one of
the following to obtain the legend organization that you want:
- use the ACROSS= option, and also set ORDER= ROWMAJOR
- use the DOWN= option, and also set ORDER=COLUMNMAJOR

A legend might be dropped if the total legend area in the graph exceeds the
percentage that is set by the MAXLEGENDAREA= option in an ODS GRAPHICS
ON statement that is in effect for the output destination. A legend might also be
dropped if DISPLAYCLIPPED= FALSE and the full legend cannot be displayed.

Examples:

Example 1: DISCRETELEGEND Statement

The following graph using the DISCRETELEGEND statement was generated by
"Example Program". It displays two discrete legends, one that shows the
confidence limits for two ellipses and a second that shows the values for a GROUP=
variable:
Example Program

```sas
proc template;
    define statgraph discretelegend;
    begingraph;
        entrytitle "Prediction Ellipses";
        layout overlayequated / equatetype=equate;
        scatterplot x=petallength y=petalwidth / group=species name="s";
        ellipse x=petallength y=petalwidth / type=predicted alpha=0.2
            name="p80" legendlabel="80%"
            outlineattrs=graphconfidence;
        ellipse x=petallength y=petalwidth / type=predicted alpha=0.05
            name="p95" legendlabel="95%"
            outlineattrs=graphconfidence2;
        discretelegend "s" / title="Species:
            discretelegend "p80" "p95" / across=1
                autoalign=(topleft) location=inside;
        endlayout;
        entryfootnote halign=left "Fisher's Iris Data";
    endgraph;
end;

proc sgrender data=sashelp.iris template=discretelegend;
run;
```
Example 2: MERGEDLEGEND Statement

The following graph using the MERGEDLEGEND statement was generated by "Example Program". In the template definition, a grouped scatter plot is overlaid with a series plot for each group, and the two plots are referenced by a single merged legend:

```
Example Program

proc sort data=sashelp.class
  out=class;
  by sex;
run;

proc template;
  define statgraph mergedLegend;
  begingraph;
  entrytitle "Linear Regression By Gender";
  layout overlay;
    scatterplot x=height y=weight / group=sex name="scat";
    regressionplot x=height y=weight / group=sex name="reg";
    mergedlegend "scat" "reg" / border=true;
  endlayout;
  endgraph;
end;

proc sgrender data=class template=mergedLegend;
run;
```
LEGENDITEM Statement

Creates the definition for a legend item that can be included in a discrete legend.

Restriction: The LEGENDITEM statement is used with the DISCRETELEGEND and MERGEDLEGEND statements only.

Requirements: The LEGENDITEM statement must be a child of the BEGINGRAPH block. It cannot be embedded in any other GTL statement block.

The LEGENDITEM statement must appear in the global definition area of the template between the BEGINGRAPH statement and the first LAYOUT statement.

Note: The LEGENDITEM statement creates the definition for a legend item that can be included in a discrete legend.

See: “DISCRETELEGEND, MERGEDLEGEND Statements” on page 1234

Syntax

LEGENDITEM TYPE=type NAME="string" <options>;

Summary of Optional Arguments

Appearance options

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
    specifies the color of the fill when TYPE= is set to FILL.

FILLDISPLAY=STANDARD | ALL | (fill-option)
    specifies whether the fill swatch for this legend item displays fill only or displays fill and outline.

FILLEDOUTLINEDMARKERS=TRUE | FALSE
    specifies whether markers are drawn with both fill and an outline.

LINEATTRS=style-element | style-element (line-options) | (line-options)
    specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE.

MARKERATTRS=style-element | style-element (marker-options) | (marker-options)
    specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
    specifies the color of the outline when TYPE= is set to FILL.

TEXT="string"
    specifies the text to display when TYPE= is set to TEXT.

TEXTATTRS=style-element | style-element (text-options) | (text-options)
    specifies the font and color attributes of the string that is specified on the TEXT= option when TYPE= is set to TEXT.

Label options

LABEL="string"
    specifies a label to be used for the legend item.

LABELATTRS
specifies the color and font attributes of the legend item’s label.

Required Arguments

**TYPE=** *FILL | MARKER | MARKERLINE | LINE | TEXT*

specifies a type for the legend item.

- **FILL**
  - specifies a fill (displayed as a filled square).
  - The appearance can be managed with the FILLATTRS= and OUTLINEATTRS= options.

- **MARKER**
  - specifies a marker.
  - The appearance can be managed with the MARKERATTRS= option.

- **MARKERLINE**
  - specifies a marker and a line.
  - The appearance can be managed with the MARKERATTRS= and LINEATTRS= options.

- **LINE**
  - specifies a line. The appearance can be managed with the LINEATTRS= option.

- **TEXT**
  - specifies text that can be displayed in the legend area.
  - The text string is defined with the TEXT= option, and the text appearance can be managed with the TEXTATTRS= option.

**NAME=** "string"

assigns a name to the legend item for reference in a DISCRETELEGEND or MERGEDLEGEND statement.

Restrictions

The string is case sensitive and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Optional Arguments

**FILLATTRS=** *style-element | style-element (fill-options) | (fill-options)*

specifies the color of the fill when TYPE= is set to FILL.

Default: The GraphDataDefault style element.

Interaction: The TRANSPARENCY attribute cannot be derived from the style element, but it can be set with this option.

See:
- “General Syntax for Attribute Options” on page 1519 for the syntax on using a *style-element.*
- “Fill Color Options” on page 1520 for available *fill-options.*

**FILLDISPLAY=** *STANDARD | ALL | (fill-option)*

specifies whether the fill swatch for this legend item displays fill only or displays fill and outline.
Note: This feature applies to SAS 9.4M3 and to later releases.

STANDARD
   ALL
       displays the fill and outline.

(fill-option)
   specifies a list of one or more fill options.
   The list can include one or both of the following options:
   FILL
       displays only the fill.
   OUTLINE
       displays only the outline.

Default      STANDARD
Requirement  You must specify TYPE=FILL for this option to have any effect.

FILLEDOUTLINEDMARKERS=TRUE | FALSE
   specifies whether markers are drawn with both fill and an outline.
   TRUE
       draws filled markers (marker symbols with the suffix FILLED) using both fill and an outline.
       Interaction  When this option is TRUE, the marker fill is drawn using the FILLATTRS= specification, and the outline is drawn using the OUTLINEATTRS= specification.
   FALSE
       draws the markers using fill or an outline, but not both.

Default      FALSE
See  "boolean " on page 1479 for other Boolean values that you can use.

LABEL="string"
   specifies a label to be used for the legend item.

Default      No label is displayed
Tip          The font and color attributes for the label are specified by the LABELATTRS= option.

LABELATTRS
   specifies the color and font attributes of the legend item’s label.

Default      The GraphValueText style element.
Interaction  For this option to take effect, the LABEL= option must also be specified.
See  "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.
     "Text Options" on page 1525 for available text-options.
LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the appearance of the line when TYPE= is set to LINE or MARKERLINE.

Default  The GraphDataDefault style element.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

MARKERATTRS=style-element | style-element (marker-options) | (marker-options)
specifies the appearance of the marker when TYPE= is set to MARKER or MARKERLINE.

Default  The GraphDataDefault style element.

Interaction  The WEIGHT attribute cannot be derived from the style element, but it can be set with this option.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Marker Options” on page 1523 for available marker-options.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the color of the outline when TYPE= is set to FILL.

Default  The GraphDataDefault style element.

Restriction  This option uses only the color specification in the style element or line options. The line pattern and line thickness specifications are ignored.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

TEXT=“string”
specifies the text to display when TYPE= is set to TEXT. The font and color attributes for the text are specified by the TEXTATTRS= option.

Default  a blank space

TEXTATTRS=style-element | style-element (text-options) | (text-options)
specifies the font and color attributes of the string that is specified on the TEXT= option when TYPE= is set to TEXT.

Default  The GraphValueText style element.

See  “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.
Details

The LEGENDITEM statement creates a definition for a legend item that can be included in a legend. The item that you define is independent of the data and enables you to customize the legend to enhance or replace a standard legend. For example, to display annotation text within the legend area, you can define a LEGENDITEM statement with TYPE=TEXT and specify the text in the TEXT= option.

As demonstrated in the "Example Program" on page 1257, multiple LEGENDITEM statements can be used to replace a plot statement's data-driven legend by defining custom legend items to display in the legend. This use enables you to communicate information that is not in the data. For this use, you would define one or more LEGENDITEM statements to specify legend attributes, and then reference those items in your legend statement. You must not reference the plot statement itself in the legend statement. Although no direct connection would exist between the plot data and the legend, you could communicate the connection by managing the visual attributes in both the plot and the legend.

The LEGENDITEM statement must be located within the BEGINGRAPH block but outside of the outermost layout block. You can use multiple LEGENDITEM statements to define multiple definitions. In that case, each definition specifies a single legend entry and each item name must be referenced in the legend statement.

Note: A single legend statement can reference multiple item names and also multiple plot names.

When specifying attribute options for a particular LEGENDITEM statement, options that do not apply to the specified TYPE= value are ignored. For example, the MARKERATTRS= option sets the attributes for a marker and is useful if you set TYPE=MARKER or TYPE=MARKERLINE. However, if TYPE= is set to a value that does not display a marker symbol, then the MARKERATTRS= option is ignored.

Example: LEGENDITEM Statement

The following graph was generated by the "Example Program" on page 1257. The example specifies three LEGENDITEM statements to define graphical properties for two marker symbols and a text string. The NAME= option in each LEGENDITEM statement assigns a name to the definition. Those names are referenced in a DISCRETELEGEND statement so that the definitions are displayed in the graph legend. To correlate the legend with the scatter plot, the example creates an attribute map that matches values M and F to the same graphical properties that are specified in the LEGENDITEM statements. That attribute map is referenced in the scatter plot.
Example Program

```plaintext
proc template;
  define statgraph scatterplot;
  begingraph;
      entrytitle "Team Tryouts: Height and Weight by Sex";
      discreteattrmap name="symbols" / ignorecase=true trimleading=true;
      value "m" / markerattrs=(color=blue symbol=diamondfilled);
      value "f" / markerattrs=(color=red symbol=circlefilled);
      enddiscreteattrmap;
      discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols";
      legendItem type=marker name="m_marker" / markerattrs=(color=blue symbol=diamondfilled) label="70% of boys present";
      legendItem type=marker name="f_marker" / markerattrs=(color=red symbol=circlefilled) label="75% of girls present";
      legendItem type=text name="status" / text="(5 health forms missing)";
      layout overlay;
      scatterplot x=height y=weight / group=groupmarkers;
      discretelegend "m_marker" "f_marker" "status" / autoitemsize=true;
  endgraph;
end;
proc sgrender data=sashelp.class template=scatterplot;
run;
```
LEGENDTEXTITEMS Statement

Creates the definition for data-driven text items that can be included in a discrete legend.

Restrictions:

- The LEGENDTEXTITEMS statement is used only with the DISCRETELEGEND statement.
- Grouping is not supported.
- The maximum number of items that the LEGENDTEXTITEMS statement can contribute to the legend is 100. If this limit is exceeded, the LEGENDTEXTITEMS statement is dropped. If no other statements contribute to the legend in that case, the legend is not drawn.

Requirements:

- The LEGENDTEXTITEMS statement must be a child of the BEGINGRAPH block. It cannot be embedded in any other GTL statement block.
- The LEGENDTEXTITEMS statement must appear in the global definition area of the template between the BEGINGRAPH statement and the first LAYOUT statement.
- The TYPE= option in the legend statement that references this statement must be set to ALL or TEXT.

Note:

- This feature applies to SAS 9.4M3 and to later releases.

See:

"DISCRETELEGEND, MERGEDLEGEND Statements" on page 1234

Syntax

```
LEGENDTEXTITEMS NAME="string" TEXT=column </options>;
```

Required Arguments

- **NAME="string"**
  assigns a name to the legend items for reference in a DISCRETELEGEND or MERGEDLEGEND statement.
  
  **Restriction** This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

- **TEXT=column**
  specifies the column in the plot data set that contains the text items.
  
  **Notes** One item is added for each observation. Grouping is not supported.
  
  The TEXT column should not contain missing values. A missing TEXT column value is treated as if no text is specified. If an observation contains a missing TEXT column value and a valid LABEL column value, only the label value is added to the legend for that observation. If both values are missing, nothing is added to the legend.
  
  If the plot data is a SAS data set, the text items are arranged in data order in the plot legend. If the plot data is a CAS in-memory table, the text items are arranged in ascending order of the text column values.
Optional Arguments

**LABEL=column**
specifies the column that contains the labels for the legend items.

- **Default**
  No labels are displayed for the items

- **Note**
  Each observation that has a LABEL column value should have a corresponding TEXT column value. If an observation contains a valid LABEL column value and a missing TEXT column value, only the label value is added to the legend for that observation.

- **Tip**
  The font and color attributes for the label are specified by the LABELATTRS= option.

**LABELATTRS=style-element | style-element (text-options) | (text-options)**
specifies the color and font attributes of the legend item labels.

- **Default**
  The GraphValueText style element.

- **Note**
  Space is reserved in the legend for the height of the legend label text regardless of whether the LABEL= option is specified.

- **Tip**
  If you are not using the LABEL= option, specify LABELATTRS=(SIZE=0pt) to reclaim the space that is reserved for the label text height.

- **See**
  “General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element value.

  “Text Options” on page 1525 for available text-options.

**TEXTATTRS=style-element | style-element (text-options) | (text-options)**
specifies the font and color attributes of the text values that is specified on the TEXT= option.

- **Default**
  The GraphValueText style element.

- **See**
  “General Syntax for Attribute Options” on page 1519 for the syntax for using a style-element value.

  “Text Options” on page 1525 for available text-options.

Details

The LEGENDTEXTITEMS statement creates a definition for data-driven text legend items that can be included in a discrete legend. The items that you define in the data are independent of the plot and enable you to customize the legend in order to enhance or replace a standard legend. The text items are stored in the plot data set in the column that is specified by the TEXT= option. The LEGENDTEXTITEMS statement must be located within the BEGINGRAPH block but outside of the outermost layout block. You reference the LEGENDTEXTITEMS statement in your legend statement by the name specified in the NAME= option. You must not reference the plot statement itself in the legend statement. To add items from multiple columns, specify one LEGENDTEXTITEMS statement for each column, and then reference all of the LEGENDTEXTITEMS statements in your legend statement. Although no direct connection would exist between the plot data and the
Example: LEGENDTEXTITEMS Statement

The following graph was generated by the “Example Program” on page 1260. The graph shows a scatter plot of height and weight by sex for 19 participants in a team tryout. To reduce clutter in the plot, a numeric ID is used to label the marker for each individual in the plot. A legend in the right margin of the graph displays the participant name for each ID.

Example Program

In this example program, a LEGENDTEXTITEMS statement is used to create the legend of IDs and names in the right margin of the graph. The name IDLEGEND is assigned to the LEGENDTEXTITEMS statement. The ID column provides the legend text items, and the Name column provides a name as the label for each text item. Here is the SAS code.

```sas
/* Assign a numeric ID to each name */
data class;
  set sashelp.class;
  id=_N_;
run;

/* Define the template for the graph */
proc template;
  define statgraph scatterplot;
  begingraph;
    entrytitle "Team Tryouts: Height and Weight by Sex";
    legendtextItems name="idlegend" text=id / label=name;
    layout lattice / columns=2 columnweights=(85 15);
    layout overlay /
xaxisopts=(griddisplay=on
    gridattrs=(color=lightgray pattern=dot))
yaxisopts=(griddisplay=on
    gridattrs=(color=lightgray pattern=dot));
scatterplot x=height y=weight / name="scatter"
    group=sex datalabel=id;
discretelegend "scatter" / location=inside
    autoalign=(bottom bottomright);
endlayout;
layout overlay;
discretelegend "idlegend" /
    title="ID/Name" titleattrs=(weight=bold)
    valign=top border=false
    order=rowmajor across=1;
endlayout;
endlayout;
endgraph;
end;

/* Render the graph */
proc sgrender data=class template=scatterplot;
run;
PART 7

Text Statements

Chapter 10
Managing Text Items .................................................. 1265

Chapter 11
Text Statements .......................................................... 1275
Managing Text Items

Overview

The ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements all use the same syntax for specifying one or more pieces of text called text items. For example, here is the ENTRYTITLE syntax:

ENTRYTITLE text-item <...<text-item>>&lt;/options&gt;;

Each text item can be specified using the following syntax:

&lt;prefix-option...&lt;prefix-option&gt;'string' | dynamic | character-expression | {text-command}

The simplest specification for each statement is to specify a string in quotation marks. For example, you might specify a graph title as follows:

entrytitle "Height and Weight by Sex";

In this example, the string "Height and Weight by Sex" is formatted as the graph title and displayed in the title position. If multiple strings are specified, then they are concatenated into a single line of text. This second specification generates the same title:
entrytitle "Height * "and " Weight " by " Sex";

To provide control over the text, multiple prefix options can precede each text item, and the text item can be a string literal, a dynamic, or a text command. All text items with the same HALIGN= setting are concatenated into one string. Up to three strings with different horizontal alignment can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

- A string must be enclosed in quotation marks.
- A character expression must be enclosed in an EVAL function.
- A text command must be enclosed in braces. (See “Using Text Commands” on page 1269).

Using Prefix Options

Available Prefix Options

The following prefix options are available on ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements:

HALIGN
specifies the horizontal alignment of a text item.

TEXTATTRS
specifies font attributes for a text item.

When used, a prefix option applies not only to immediately following piece of text but also to ALL subsequent text strings and text-commands. If the same prefix option appears more than once, then each use overrides the last used prefix of the same name.

Managing Horizontal Alignment

For the ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements, the default horizontal alignment is CENTER.

| One Two Three Four Five Six |

entry "One" " Two" " Three" " Four" " Five" " Six";

The HALIGN= option can be used to change the horizontal alignment to LEFT, CENTER, or RIGHT. The following specification left-justifies the text:

| One Two Three Four Five Six |

entry halign=left
Text items are positionally concatenated by alignment area. For example, the following specification left-justifies the first three strings and right-justifies the last three strings:

```
One Two Three    Four Five Six
```

even if the HALIGN= specifications are jumbled, the final text is nevertheless positionally concatenated by alignment area:

```
One Two       Three Four       Five Six
```

Note: When long strings are used or the bounding container is constrained, the alignment areas might overlap.

Managing Font Attributes

For the ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements, the default font attributes are determined by the active ODS style. The TEXTATTRS= option can be used to change the default font attributes. The TEXTATTRS= option syntax is as follows (see "General Syntax for Attribute Options" on page 1519 for the syntax on using a style element and "Text Options" on page 1525 for available text options):

```
TEXTATTRS=style-element | style-element(text-options) | (text-options)
```

For example, the following ENTRYTITLE statement uses prefix options to create this title line:

```
Left side        Center Text        Right side
```

- The TEXTATTRS= options are reset each time a new TEXTATTRS= appears—there is no "carry over" or accumulation of the settings.
- Blanks (spaces) must be provided as needed to achieve the desired concatenation. When concatenating dynamic variables that are stripped of
leading and trailing blanks, a literal space must be inserted to separate them, as shown in the following example:

```
entry _DYN1 " " _DYN2;
```

Alternatively, font attributes can be specified by overriding the style element defaults. The following specification overrides the settings of the GraphTitleText style element, which sets the default attributes for ENTRYTITLE text:

```
entrytitle
  halign=left
textattrs=GraphTitleText(color=black style=italic) "Left"
textattrs=GraphTitleText(color=red style=italic) " side"
  halign=center
textattrs=(color=black) "Center"
textattrs=(color=red) " Text"
  halign=right
textattrs=GraphTitleText(color=black weight=bold) "Right "
textattrs=GraphTitleText(color=red weight=bold) "side";
```

Dynamic variables can also be used in the text strings. In the following ENTRYTITLE statement, assume that _DEPLABEL and _MODELLABEL are dynamic variables that are specified on PROC TEMPLATE’s DYNAMIC statement:

```
entrytitle "Residual by Predicted for " _DEPLABEL
  halign=left textattrs=GraphTitleText _MODELLABEL /
    pad=(bottom=5);
```

- The default style element for ENTRYTITLE is GraphTitleText, so all three text items (one literal and two dynamic variables) get these font properties as a starting point.
- The text "Residual by Predicted for " _DEPLABEL is center-aligned by default.
- The prefix options HALIGN= and TEXTATTRS= override the center alignment and font properties for the text _MODELLABEL.

This could have been coded as follows:

```
entrytitle halign=left
  textattrs=GraphTitleText _MODELLABEL
  halign=center textattrs() 
    "Residual by Predicted for " _DEPLABEL;
```

In this example, the second HALIGN= and TEXTATTRS() are necessary to reset alignment and font properties to their defaults.

The string length of the resolved dynamic _MODELLABEL does not affect the placement of the center-aligned text.
Using Text Commands

Available Text Commands

Text commands on ENTRY, ENTRYTITLE, and ENTRYFOOTNOTE statements are special in-line instructions that either modify the appearance of the text or script special characters. The following text commands are available:

{SUB}
specifies that the string or dynamic appears as a subscript.

{SUP}
specifies that the string or dynamic appears as a superscript.

{UNICODE}
specifies a glyph (graphical character) to be displayed using its Unicode specification or keyword equivalent.

The general form of a text command is

{command argument(s)}

The opening and closing braces are required to denote the scope of the command.

Subscripting and Superscripting Text

The {SUB} and {SUP} text commands are used to subscript and superscript text. Each of these text commands accepts a string or a dynamic for its argument(s).

In the following example, if _RSQUARE is an existing dynamic that resolves to the value 0.7434, then the following ENTRY specification superscripts the string value "2" to generate this text:

\[ R^2 = 0.7434 \]

entry textattrs=(weight=bold) "R" {sup "2"}
   textattrs=() *=* _RSQUARE;

The textattrs=() option cancels all style overrides and reverts to the default text properties.

Using UNICODE Text

The {UNICODE} text command places special characters into the text and accepts any of the following for its argument(s):

- a hexadecimal Unicode Code Point for a character (for example, "03B1"x)
- a reserved keyword for a commonly used code point (for example, BETA)
- a dynamic that resolves to a hexadecimal value or keyword.

The tables in "Reserved Keywords and Unicode Values" on page 1271 provide a list of the commonly used reserved keywords and Unicode values (the tables are not complete, but they provide an idea about what is possible).

Multiple arguments can be used within the scope of a single UNICODE text command. For example, the following two specifications are equivalent:

```
{unicode "03b1"x beta}
{unicode "03b1"x} {unicode beta}
```

The default formatting for the UNICODE text is derived from the GraphUnicodeText style element.

In the following example, if _ALPHAVAL is an existing dynamic that resolves to the value 0.05, then the following ENTRY specification generates this text:

```plaintext
\alpha = 0.05
```

```
entry {unicode alpha} " = " _ALPHAVAL;
```

By combining the TEXTATTRS= prefix option with the {SUB} and {UNICODE} text commands, you can generate the following text:

```
E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2
```

```
Entry textattrs=(style=italic) "E(Y)" textattrs=() " = 
{unicode beta} {sub "0"} * + *
{unicode beta} {sub "1"} *x* {sub "1"}*
" + " {unicode beta} {sub "2"} "x" {sub "2"};
```

### Rules for Unicode and Special Character Specifications

The following rules apply to Unicode and special character specifications in ODS graphics:

- Each character can be specified by looking up its code and specifying it as a hexadecimal constant. Example: `{unicode ‘221e’x}`.
- Lowercase Greek letters can be specified by using names instead of hexadecimal constants. Example: `{unicode alpha}`.
- Uppercase Greek letters can be specified by using names followed by _u instead of a hexadecimal constant. Example: `{unicode alpha_u}`.
- Superscript and subscript have special abbreviations. Examples: `{sup 2}` and `{sub 2}`.
- The {SUP} and {SUB} specifications must not appear escaped and in quotation marks in GTL. They must appear outside of quotation marks.
Some characters overprint the character that comes before. Example: ‘El nin’ {tilde} ‘o’, which is equivalent to ‘El nin’ {unicode ‘0303’x} ‘o’ creates ‘El niño’.

Specifications inside quotation marks are escaped. Example: "{*ESC*}{unicode beta}".

Specifications outside quotation marks are not escaped. Example: {unicode beta}.

For more information about using text throughout GTL (for example, using Unicode values in labels), see SAS Graph Template Language: User’s Guide.

---

Reserved Keywords and Unicode Values

Overview

The tables in this section show some of the reserved keywords and Unicode values that can be used with the UNICODE text command. For information about rendering Unicode characters, see "Managing the String on Text Statements" in SAS Graph Template Language: User’s Guide.

Note the following:

- Keywords and Unicode values are not case-sensitive: "03B1"x is the same code point as "03b1"x.
- The word blank is the keyword for a blank space.

Lowercase Greek Letters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>α</td>
<td>03B1</td>
<td>lowercase alpha</td>
</tr>
<tr>
<td>beta</td>
<td>β</td>
<td>03B2</td>
<td>lowercase beta</td>
</tr>
<tr>
<td>gamma</td>
<td>γ</td>
<td>03B3</td>
<td>lowercase gamma</td>
</tr>
<tr>
<td>delta</td>
<td>δ</td>
<td>03B4</td>
<td>lowercase delta</td>
</tr>
<tr>
<td>epsilon</td>
<td>ε</td>
<td>03B5</td>
<td>lowercase epsilon</td>
</tr>
<tr>
<td>zeta</td>
<td>ζ</td>
<td>03B6</td>
<td>lowercase zeta</td>
</tr>
<tr>
<td>eta</td>
<td>η</td>
<td>03B7</td>
<td>lowercase eta</td>
</tr>
</tbody>
</table>
### Lowercase Greek Letters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>theta</td>
<td>θ</td>
<td>03B8</td>
<td>lowercase theta</td>
</tr>
<tr>
<td>iota</td>
<td>ι</td>
<td>03B9</td>
<td>lowercase iota</td>
</tr>
<tr>
<td>kappa</td>
<td>κ</td>
<td>03BA</td>
<td>lowercase kappa</td>
</tr>
<tr>
<td>lambda</td>
<td>λ</td>
<td>03BB</td>
<td>lowercase lambda</td>
</tr>
<tr>
<td>mu</td>
<td>μ</td>
<td>03BC</td>
<td>lowercase mu</td>
</tr>
<tr>
<td>nu</td>
<td>ν</td>
<td>03BD</td>
<td>lowercase nu</td>
</tr>
<tr>
<td>xi</td>
<td>ξ</td>
<td>03BE</td>
<td>lowercase xi</td>
</tr>
<tr>
<td>omicron</td>
<td>ο</td>
<td>03BF</td>
<td>lowercase omicron</td>
</tr>
<tr>
<td>pi</td>
<td>π</td>
<td>03C0</td>
<td>lowercase pi</td>
</tr>
<tr>
<td>rho</td>
<td>ρ</td>
<td>03C1</td>
<td>lowercase rho</td>
</tr>
<tr>
<td>sigma</td>
<td>σ</td>
<td>03C3</td>
<td>lowercase sigma</td>
</tr>
<tr>
<td>tau</td>
<td>τ</td>
<td>03C4</td>
<td>lowercase tau</td>
</tr>
<tr>
<td>upsilon</td>
<td>υ</td>
<td>03C5</td>
<td>lowercase upsilon</td>
</tr>
<tr>
<td>phi</td>
<td>φ</td>
<td>03C6</td>
<td>lowercase phi</td>
</tr>
<tr>
<td>chi</td>
<td>χ</td>
<td>03C7</td>
<td>lowercase chi</td>
</tr>
<tr>
<td>psi</td>
<td>ψ</td>
<td>03C8</td>
<td>lowercase psi</td>
</tr>
<tr>
<td>omega</td>
<td>ω</td>
<td>03C9</td>
<td>lowercase omega</td>
</tr>
</tbody>
</table>

### Uppercase Greek Letters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha_u</td>
<td>Α</td>
<td>0391</td>
<td>uppercase alpha</td>
</tr>
<tr>
<td>beta_u</td>
<td>Β</td>
<td>0392</td>
<td>uppercase beta</td>
</tr>
<tr>
<td>gamma_u</td>
<td>Γ</td>
<td>0393</td>
<td>uppercase gamma</td>
</tr>
<tr>
<td>Keyword</td>
<td>Glyph</td>
<td>Unicode</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>---------</td>
<td>-------------------</td>
</tr>
<tr>
<td>delta_u</td>
<td>Δ</td>
<td>0394</td>
<td>uppercase delta</td>
</tr>
<tr>
<td>epsilon_u</td>
<td>Ε</td>
<td>0395</td>
<td>uppercase epsilon</td>
</tr>
<tr>
<td>zeta_u</td>
<td>Ζ</td>
<td>0396</td>
<td>uppercase zeta</td>
</tr>
<tr>
<td>eta_u</td>
<td>Η</td>
<td>0397</td>
<td>uppercase eta</td>
</tr>
<tr>
<td>theta_u</td>
<td>Θ</td>
<td>0398</td>
<td>uppercase theta</td>
</tr>
<tr>
<td>iota_u</td>
<td>Ι</td>
<td>0399</td>
<td>uppercase iota</td>
</tr>
<tr>
<td>kappa_u</td>
<td>Κ</td>
<td>039A</td>
<td>uppercase kappa</td>
</tr>
<tr>
<td>lambda_u</td>
<td>Λ</td>
<td>039B</td>
<td>uppercase lambda</td>
</tr>
<tr>
<td>mu_u</td>
<td>Μ</td>
<td>039C</td>
<td>uppercase mu</td>
</tr>
<tr>
<td>nu_u</td>
<td>N</td>
<td>039D</td>
<td>uppercase nu</td>
</tr>
<tr>
<td>xi_u</td>
<td>Ξ</td>
<td>039E</td>
<td>uppercase xi</td>
</tr>
<tr>
<td>omicron_u</td>
<td>Ο</td>
<td>039F</td>
<td>uppercase omicron</td>
</tr>
<tr>
<td>pi_u</td>
<td>Π</td>
<td>03A0</td>
<td>uppercase pi</td>
</tr>
<tr>
<td>rho_u</td>
<td>Ρ</td>
<td>03A1</td>
<td>uppercase rho</td>
</tr>
<tr>
<td>sigma_u</td>
<td>Σ</td>
<td>03A3</td>
<td>uppercase sigma</td>
</tr>
<tr>
<td>tau_u</td>
<td>Τ</td>
<td>03A4</td>
<td>uppercase theta</td>
</tr>
<tr>
<td>upsilon_u</td>
<td>Υ</td>
<td>03A5</td>
<td>uppercase upsilon</td>
</tr>
<tr>
<td>phi_u</td>
<td>Φ</td>
<td>03A6</td>
<td>uppercase phi</td>
</tr>
<tr>
<td>chi_u</td>
<td>Χ</td>
<td>03A7</td>
<td>uppercase chi</td>
</tr>
<tr>
<td>psi_u</td>
<td>Ψ</td>
<td>03A8</td>
<td>uppercase psi</td>
</tr>
<tr>
<td>omega_u</td>
<td>Ω</td>
<td>03A9</td>
<td>uppercase omega</td>
</tr>
</tbody>
</table>
### Special Characters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prime</td>
<td>´</td>
<td>00B4</td>
<td>single prime sign</td>
</tr>
<tr>
<td>bar</td>
<td>̅</td>
<td>0305</td>
<td>combining overline¹</td>
</tr>
<tr>
<td>bar2</td>
<td>̿</td>
<td>033F</td>
<td>combining double overline¹</td>
</tr>
<tr>
<td>tildes</td>
<td>̃</td>
<td>0303</td>
<td>combining tilde¹</td>
</tr>
<tr>
<td>hats</td>
<td>^</td>
<td>0302</td>
<td>combining circumflex accent¹</td>
</tr>
</tbody>
</table>

¹ This is an overstriking character that requires a Unicode font to render properly.
Text Statements

Dictionary

ENTRY Statement

Displays a line of text in the graphical area.

Requirement: An ENTRY statement must be specified within a LAYOUT, HEADER, SIDEBAR, or CELL statement block.

Syntax

ENTRY text-item <text-item …> <options>;

Summary of Optional Arguments

Appearance options

  BACKGROUNDCOLOR=style-reference | color
  specifies the color of the text background.

  BORDER=TRUE | FALSE
  specifies whether a border is displayed around the text.

  BORDERATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the border line around the text.

  OPAQUE=TRUE | FALSE
  specifies whether the text background is opaque (TRUE) or transparent (FALSE).

  OUTERPAD=AUTO | dimension | (pad-options)
  specifies the amount of extra space to add outside the entry border.

  PAD=dimension | (pad-options)
  specifies the amount of extra space that is reserved inside the entry border.
ROTATE=0 | 90 | 180 | 270
specifies the angle of text rotation measured in degrees.

TEXTATTRS=style-element | style-element (text-options) | (text-options)
as a statement option, specifies the properties of the text. As a prefix-option, specifies the properties of individual text-items.

Location options
AUTOALIGN=NONE | AUTO | (location-list)
specifies whether the text is automatically aligned within its parent when nested within an overlay-type layout.

VALIGN=CENTER | TOP | BOTTOM
specifies the vertical alignment of the text.

Text options
HALIGN=CENTER | LEFT | RIGHT
as a statement option, specifies the horizontal alignment of the text. As a prefix-option, specifies the horizontal alignment of individual text-items.

Required Argument
<text-item> <...<text-item>>
specifies one or more pieces of text for the entry. Each text-item has the following form:

<prefix-option> "string" | dynamic | character-expression | {text-command}

Each piece of text can have multiple prefix settings that precede the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All text-items with the same HALIGN= setting are concatenated into one string. Up to three strings can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

When a prefix option is used, it applies not only to the text that immediately follows the prefix option, but also to all subsequent text strings and text commands. If the same prefix option appears more than once, then that prefix overrides the last used prefix of the same name for the subsequent text strings and text commands.

Restriction
The maximum length for the entry text is 512 characters in SAS 9.4 and earlier releases. Starting with SAS 9.4M1, this restriction is removed.

Requirements
string must be enclosed in quotation marks.

class-expression must be enclosed in an EVAL function.

text-command must be enclosed in braces.

Note
Leading spaces are preserved, and trailing spaces are stripped.

See
Chapter 10, “Managing Text Items,” on page 1265
Optional Arguments

**AUTOALIGN=NONE | AUTO | (location-list)**

specifies whether the text is automatically aligned within its parent when nested within an overlay-type layout. For more information about how child positions are determined in an overlay-type layout, see the “LAYOUT OVERLAY Statement” on page 149.

**NONE**

 does not automatically align the text within the area.

Alignment is set with **HALIGN**= and **VALIGN**= options.

**AUTO**

within the parent layout, attempts to center the text in the area that is farthest from any surrounding data point markers.

**Restriction**

This option is available only if the parent layout contains a scatter plot. Otherwise, it is ignored.

**(location-list)**

within the available area, restricts the text's possible locations to those locations in the specified **location-list**, and uses the **location-list** position that least collides with the other graphics features in the area.

The **location-list** is space-separated and can contain any of these locations: TOPLEFT, TOP, TOPRIGHT, LEFT, CENTER, RIGHT, BOTTOMLEFT, BOTTOM, and BOTTOMRIGHT.

Example: (TOPRIGHT TOPLEFT)

**Default**

NONE

**Requirement**

For this option to take effect, the ENTRY statement must be in a LAYOUT OVERLAY or LAYOUTOVERLAYEQUATED block.

**Interaction**

When AUTOALIGN= is not NONE and the parent layout is an overlay-type layout, the **HALIGN**= and the **VALIGN**= options are ignored.

**BACKGROUNDCOLOR=style-reference | color**

specifies the color of the text background.

**style-reference**

specifies a style reference in the form **style-element**: **style-attribute**.

Only the **style-attribute** named COLOR or CONTRASTCOLOR is used.

**Default**

The GraphBackground:Color style reference.

**Interaction**

**OPAQUE**= TRUE must be in effect in order for the color to be seen. By default, **OPAQUE**=FALSE.

**BORDER=TRUE | FALSE**

specifies whether a border is displayed around the text.

**Default**

FALSE

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=style-element | style-element (line-options) | (line-options)**

specifies the attributes of the border line around the text.
The GraphBorderLines style element.

**Interaction**

BORDER= TRUE must be set for this option to have any effect.

**See**

"General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Line Options" on page 1522 for available line-options.

**HALIGN=** CENTER | LEFT | RIGHT  
as a statement option, specifies the horizontal alignment of the text. As a prefix-option, specifies the horizontal alignment of individual text-items. Each text-item has a horizontal alignment, and text-items with the same alignment are always grouped together.

**Default** CENTER

**Interaction**

This option is ignored when AUTOALIGN= is not NONE and the parent layout is an overlay-type layout.

**OPAQUE=** TRUE | FALSE  
specifies whether the text background is opaque (TRUE) or transparent (FALSE).

**Default** FALSE

**Interaction**

When this option is set to FALSE, the background color is not used.

**See**

“boolean” on page 1479 for other Boolean values that you can use.

**OUTERPAD=** AUTO | dimension | (pad-options)  
specifies the amount of extra space to add outside the entry border.

**AUTO**

specifies that the default outside padding for this component is used.

**dimension**

specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

**(pad-options)**

a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can include one or more of the following pad options:

**LEFT=** dimension  
specifies the amount of extra space to add to the left side.

**RIGHT=** dimension  
specifies the amount of extra space to add to the right side.

**TOP=** dimension  
specifies the amount of extra space to add to the top.

**BOTTOM=** dimension  
specifies the amount of extra space to add to the bottom.

**Note**

Sides that are not assigned padding are padded with the default amount.
Tip Use *pad-options* to create non-uniform padding.

**Default** AUTO

**Note** The default units for *dimension* are pixels.

**See** “*dimension*” on page 1480

**PAD=*dimension* | (pad-options)**
specifies the amount of extra space that is reserved inside the entry border.

**Dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

**(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can include one or more of the following pad options:

- **LEFT=*dimension***
specifies the amount of extra space to add to the left side.

- **RIGHT=*dimension***
specifies the amount of extra space to add to the right side.

- **TOP=*dimension***
specifies the amount of extra space to add to the top.

- **BOTTOM=*dimension***
specifies the amount of extra space to add to the bottom.

**Note** Sides that are not assigned padding are padded with the default amount.

**Tip** Use *pad-options* to create non-uniform padding.

**Defaults** Prior to SAS 9.4M6, padding is LEFT=3, RIGHT=3, TOP=0, and BOTTOM=0.

Starting with SAS 9.4M6, if **ROTATE=0** on page 1279, the default padding is LEFT=3, RIGHT=3, TOP=0, and BOTTOM=0. If **ROTATE** is other than 0, the default padding is LEFT=0, RIGHT=0, TOP=3, and BOTTOM=3.

**Note** The default units for *dimension* are pixels.

**See** “*dimension*” on page 1480

**ROTATE=0 | 90 | 180 | 270**
specifies the angle of text rotation measured in degrees. The angle is measured from a horizontal line passing through the middle of the bounding box of the text, counter-clockwise starting at the center of the bounding box.

**Default** 0. No rotation is performed.

**Restriction** Only angles of 0, 90, 180, or 270 degrees are allowed.
The bounding box is determined by the size of the text in the current font plus any horizontal and vertical padding. See `TEXTATTRS=` and `PAD=`.

**TEXTATTRS=**

- **style-element | style-element (text-options) | (text-options)**
  - as a *statement option*, specifies the properties of the text. As a *prefix-option*, specifies the properties of individual *text-items*.

**Default**

The GraphValueText style element.

**Notes**

- This option can be used as both a prefix option and a statement option. When used as a prefix option, it overrides the statement option.
- When used as a prefix option, `TEXTATTRS=` cancels the last used `TEXTATTRS=` prefix option. It resets all text options to those set by 1) the `TEXTATTRS=` statement option or 2) the default style element for the statement (GraphValueText) if no `TEXTATTRS=` statement option is used.

**See**

- "General Syntax for Attribute Options" on page 1519 for the syntax on using a *style-element*.
- "Text Options" on page 1525 for available *text-options*.

**VALIGN=**

- **CENTER | TOP | BOTTOM**
  - specifies the vertical alignment of the text.

**Default**

CENTER

**Interaction**

- This option is ignored when `AUTOALIGN=` is not `NONE` and the parent layout is an overlay-type layout.

**Prefix Options**

**HALIGN=**

- **CENTER | LEFT | RIGHT**
  - specifies the horizontal alignment of the text that follows.

**See**

- `HALIGN=` on page 1278

**TEXTATTRS=**

- **style-element | style-element (text-options) | (text-options)**
  - specifies the properties of the text that follows.

**See**

- `TEXTATTRS=` on page 1280

**Text Commands**

- `{ SUB "string" | dynamic }`  
  - *text-command* that specifies that the string or dynamic is to appear as subscript text.

**See**

- "Rules for Unicode and Special Character Specifications" on page 1270

**Example**

entry "y = " b{sub "0"} " + b" {sub "1"} "x";
\{ \textbf{SUP} \ "string" \mid \textbf{dynamic} \} \\
\textit{text-command} that specifies that the string or dynamic is to appear as superscript text.

\begin{itemize}
  \item See \textit{"Rules for Unicode and Special Character Specifications" on page 1270}
  \item Example \texttt{entry \"R\" \{sup \"2\}\ = \" \{format (6.4) RSQUARED\} ;}
\end{itemize}

\{ \textbf{UNICODE} \ "hex-string"x \mid \textbf{keyword} \mid \textbf{dynamic} \} \\
\textit{text-command} that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.

\begin{itemize}
  \item \textit{"hex-string"x} \\
    a four-byte hexadecimal constant that represents a UNICODE character in the current font.
    \begin{itemize}
      \item For a complete listing, see http://unicode.org/charts/charindex.html.
    \end{itemize}
  \item \textit{keyword} \\
    a SAS keyword for a UNICODE character.
    \begin{itemize}
      \item For a listing of keywords supplied by SAS, see \textit{"Reserved Keywords and Unicode Values" on page 1271}.
    \end{itemize}
  \item \textit{dynamic} \\
    a dynamic variable that resolves to either \textit{"hex-string"x} or a keyword for a UNICODE character.
\end{itemize}

\begin{itemize}
  \item Note The UNICODE text command attempts to access a UNICODE value in the current font. Not all fonts support accessing characters through their UNICODE value. Some fonts support only a limited set of UNICODE values. If the UNICODE value is not accessible, then the command might be ignored or a nonprintable character might be substituted.
  \item See \textit{"Using UNICODE Text" on page 1269}
  \item \textit{"Rules for Unicode and Special Character Specifications" on page 1270}
  \item Example \texttt{entry \{unicode alpha\} \"=" CONF;}
    \begin{itemize}
      \item \texttt{entry \{unicode \"03B1\"x\} \"=" CONF;}
    \end{itemize}
\end{itemize}

\section*{Details}

An \texttt{ENTRY} statement creates one line of text in the plot area. The statement must be specified within a LAYOUT, HEADER, SIDEBAR, or CELL statement block. It cannot be specified outside of one of these blocks, where global statements like \texttt{ENTRYTITLE} and \texttt{ENTRYFOOTNOTE} are used.
The text line specified in an ENTRY statement can be made of several pieces of the text called text-items. Statement options that are used establish properties for the entire line of text (that is, all text-items). These properties can be overridden with prefix-options that are specific to one or more text-items. See “Required Argument” on page 1276 for more information.

You can specify an ENTRY statement inside or outside of a nested statement block:

- When you specify an ENTRY statement inside a nested LAYOUT, HEADER, SIDEBAR, or CELL statement block, then, by default, the text is placed inside the bounding area of the results that the nested statement block creates.
- When you specify an ENTRY statement outside of a nested LAYOUT, HEADER, SIDEBAR, or CELL statement block, then the text is placed outside of the bounding area of the results that the nested statement block creates.

Example: ENTRY Statement

The following graph was generated by the “Example Program” on page 1282:

![Example Graph](image)

Example Program

```plaintext
proc template;
define statgraph entry;
begingraph;
layout overlay;

    entry halign=right "First entry statement" / valign=top;

    histogram weight;
```

```
entry halign=right "Second entry statement";

entry halign=right "Third entry statement" / valign=bottom pad=(bottom=40px);

endlayout;
endgraph;
end;
run;

proc sgrender data=sashelp.cars template=entry;
run;

---

**ENTRYFOOTNOTE Statement**

Displays a footnote.

**Requirement:**
You must place the ENTRYFOOTNOTE statement directly inside the BEGINGRAPH block. Do not embed it in any other GTL statement block.

**Syntax**

```
ENTRYFOOTNOTE text-item <text-item ...> </options>;
```

**Summary of Optional Arguments**

**Appearance options**

- **BACKGROUNDCOLOR=** `style-reference | color`
  specifies the color of the text background.

- **BORDER=** `TRUE | FALSE`
  specifies whether a border is displayed around the text.

- **BORDERATTRS=** `style-element | style-element (line-options) | (line-options)`
  specifies the attributes of the border line around the text.

- **OPAQUE=** `TRUE | FALSE`
  specifies whether the text background is opaque (TRUE) or transparent (FALSE).

- **OUTERPAD=** `AUTO | dimension | (pad-options)`
  specifies the amount of extra space to add outside the entry border.

- **PAD=** `dimension | (pad-options)`
  specifies the amount of extra space that is reserved inside the entry border.

- **TEXTATTRS=** `style-element | style-element (text-options) | (text-options)`
  as a statement option, specifies the properties of the text. As a prefix-option, specifies the properties of individual text-items.

**Text options**

- **HALIGN=** `CENTER | LEFT | RIGHT`
  as a statement option, specifies the horizontal alignment of the text. As a prefix-option, specifies the horizontal alignment of individual text-items.

- **HALIGNCENTER=** `AUTO | GRAPH`
specifies whether the text is centered automatically by the system or is always centered in the graph area.

SHORTTEXT=(text-item <...text-item>)
specifies alternate text to use if the specified text is too long for the output width.

TEXTFITPOLICY=WRAP | SHORT | TRUNCATE
specifies how to handle text that is too long to fit in the output width.

Required Argument

text-item <...text-item>
specifies one or more pieces of text for the footnote. Each text-item has the following form:

<prefix-option ...<prefix-option>>"string" | dynamic | character-expression | {text-command}

Each piece of text can have multiple prefix options that precede the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All text-items with the same HALIGN= setting are concatenated into one string. Up to three strings can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

When a prefix option is used, it applies not only to the text that immediately follows the prefix option, but also to all subsequent text strings and text commands. If the same prefix option appears more than once, then that prefix overrides the last used prefix of the same name for the subsequent text strings and text commands.

Restriction The maximum length for the footnote text is 512 characters in SAS 9.4 and earlier releases. Starting with SAS 9.4M1, this restriction is removed.

Requirements string must be enclosed in quotation marks.

character-expression must be enclosed in an EVAL function.

text-command must be enclosed in braces.

Note Leading spaces are preserved, and trailing spaces are stripped.

See Chapter 10, "Managing Text Items," on page 1265

Optional Arguments

BACKGROUNDCOLOR=style-reference | color
specifies the color of the text background.

style-reference
specifies a style reference in the form style-element:style-attribute.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

Default The GraphBackground:Color style reference.

Interaction OPAQUE= TRUE must be in effect in order for the color to be seen. By default, OPAQUE=FALSE.
BORDER=TRUE | FALSE
specifies whether a border is displayed around the text.

Default   FALSE

See   “boolean” on page 1479 for other Boolean values that you can use.

BORDERATTRS=style-element | style-element (line-options) | (line-options)
specifies the attributes of the border line around the text.

Default   The GraphBorderLines style element.

Interaction   BORDER= TRUE must be set for this option to have any effect.

See   “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

HALIGN=CENTER | LEFT | RIGHT
as a statement option, specifies the horizontal alignment of the text. As a prefix-option, specifies the horizontal alignment of individual text-items. Each text-item has a horizontal alignment, and text-items with the same alignment are always grouped together.

Default   CENTER

HALIGNCENTER=AUTO | GRAPH
specifies whether the text is centered automatically by the system or is always centered in the graph area.

AUTO
specifies that the system determines how the text is centered.

The text is centered as follows:

- For LAYOUT GRIDDED, LAYOUT OVERLAY3D, and LAYOUT REGION layouts, and for LAYOUT LATTICE, LAYOUT DATAPANEL, and LAYOUT DATALATTICE layouts that have more than one column, center the text in the graph area.
- For LAYOUT OVERLAY and LAYOUT OVERLAYEQUATED layouts, and for LAYOUT LATTICE, LAYOUT DATAPANEL, and LAYOUT DATALATTICE layouts that have only one column, center the text in the graph wall area. If the length of the text exceeds the width of the graph wall area, then center the text in the graph area instead.

GRAPH
specifies that the text is always centered in the graph area.

Default   AUTO

Interaction   The prefix option HALIGN= must specify CENTER for this option to have any effect.

OPAQUE=TRUE | FALSE
specifies whether the text background is opaque (TRUE) or transparent (FALSE).

Default   FALSE
When this option is set to FALSE, the background color is not used.

See “boolean” on page 1479 for other Boolean values that you can use.

**OUTERPAD=**AUTO | dimension | (pad-options)

specifies the amount of extra space to add outside the entry border.

- **AUTO**
  - specifies that the default outside padding for this component is used.

- **dimension**
  - specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

- **(pad-options)**
  - a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

  The list can include one or more of the following pad options:

  - **LEFT=**dimension
    - specifies the amount of extra space to add to the left side.

  - **RIGHT=**dimension
    - specifies the amount of extra space to add to the right side.

  - **TOP=**dimension
    - specifies the amount of extra space to add to the top.

  - **BOTTOM=**dimension
    - specifies the amount of extra space to add to the bottom.

- **Note**
  - Sides that are not assigned padding are padded with the default amount.

- **Tip**
  - Use pad-options to create non-uniform padding.

**Default**

AUTO

**Note**

The default units for dimension are pixels.

**See**

“dimension” on page 1480

**PAD=**dimension | (pad-options)

specifies the amount of extra space that is reserved inside the entry border.

- **dimension**
  - specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

- **(pad-options)**
  - a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

  The list can include one or more of the following pad options:

  - **LEFT=**dimension
    - specifies the amount of extra space to add to the left side.

  - **RIGHT=**dimension
    - specifies the amount of extra space to add to the right side.
TOP=\textit{dimension}  
        specifies the amount of extra space to add to the top.

BOTTOM=\textit{dimension}  
        specifies the amount of extra space to add to the bottom.

\textbf{Note}  
        Sides that are not assigned padding are padded with the default amount.

\textbf{Tip}  
        Use \textit{pad-options} to create non-uniform padding.

\textbf{Default}  
        LEFT=3, RIGHT=3, TOP=0, and BOTTOM=0

\textbf{Note}  
        The default units for \textit{dimension} are pixels.

\textbf{See}  
        “dimension” on page 1480

\textbf{SHORTTEXT=(text-item <...text-item>)}  
        specifies alternate text to use if the specified text is too long for the output width.  
        If the shortened text is itself too long, then it is truncated.

\textbf{Interactions}  
        This option has no effect unless \textit{TEXTFITPOLICY= SHORT}.

        This option is ignored if any \textit{text-items} include an \textit{HALIGN=} prefix option.

        The horizontal alignment of the shortened text is derived from the horizontal alignment of the text to be shortened.

\textbf{TEXTATTRS=style-element | style-element (text-options) | (text-options)}  
        as a \textit{statement option}, specifies the properties of the text.  
        As a \textit{prefix-option}, specifies the properties of individual \textit{text-items}.

\textbf{Default}  
        The GraphFootnoteText style element.

\textbf{Notes}  
        This option can be used as both a prefix option and a statement option.  
        When used as a prefix option, it overrides the statement option.

When used as a prefix option, TEXTATTRS= cancels the last used TEXTATTRS= prefix option.  
It resets all text options to those set by 1) the TEXTATTRS= statement option or 2) the default style element for the statement (GraphValueText) if no TEXTATTRS= statement option is used.

\textbf{See}  
        “General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}.

        “Text Options” on page 1525 for available \textit{text-options}.

\textbf{TEXTFITPOLICY=WRAP | SHORT | TRUNCATE}  
        specifies how to handle text that is too long to fit in the output width.

\textbf{WRAP}  
        specifies that the text wraps to the next line(s).

\textbf{Restriction}  
        Text wrapping is available only for ENTRYFOOTNOTE statements that appear in the outermost layout.
SHORT
specifies that the text indicated by the SHORTTEXT= option be substituted when the text does not fit.

Requirement You must specify the SHORTTEXT= option for this option to take effect.

Note If the short text is also too long, then it is truncated.

See “SHORTTEXT=(text-item <...text-item>)” on page 1287

TRUNCATE
specifies that the text is truncated to make it fit.

Default WRAP

Prefix Options
HALIGN= CENTER | LEFT | RIGHT
specifies the horizontal alignment of the text that follows.

See HALIGN= on page 1285

TEXTATTRS= style-element | style-element (text-options) | (text-options)
specifies the properties of the text that follows.

See TEXTATTRS= on page 1287

Text Commands
{ SUB "string" | dynamic }
text-command that specifies that the string or dynamic is to appear as subscript text.

See “Rules for Unicode and Special Character Specifications” on page 1270

Example entryfootnote "y =  b_{sub "0"} " + b" {sub "1"} "x";

{ SUP "string" | dynamic }
text-command that specifies that the string or dynamic is to appear as superscript text.

See “Rules for Unicode and Special Character Specifications” on page 1270

Example entryfootnote "R_{sup "2"} " = " \text{RSQUARED};

{ UNICODE "hex-string"x | keyword | dynamic }
text-command that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.

"hex-string"x
a four-byte hexadecimal constant that represents a UNICODE character in the current font.

For a complete listing, see http://unicode.org/charts/charindex.html.
**keyword**

A SAS keyword for a UNICODE character.

For a listing of keywords supplied by SAS, see "Reserved Keywords and Unicode Values" on page 1271.

**dynamic**

A dynamic variable that resolves to either "hex-string"x or a keyword for a UNICODE character.

**Note**

The UNICODE text command attempts to access a UNICODE value in the current font. Not all fonts support accessing characters through their UNICODE value. Some fonts support only a limited set of UNICODE values. If the UNICODE value is not accessible, then the command might be ignored or a nonprintable character might be substituted.

**See**

"Using UNICODE Text" on page 1269

"Rules for Unicode and Special Character Specifications" on page 1270

**Example**

```plaintext
entryfootnote {unicode alpha} "=" CONF;
entryfootnote {unicode "03B1"x} "=" CONF;
```

**Details**

The ENTRYFOOTNOTE statement places footnote text below the graphical area. More than one ENTRYFOOTNOTE statement can be used. Footnotes appear in the order of the ENTRYFOOTNOTE statements within the template.

---

**When you add an ENTRYFOOTNOTE statement to a template definition, place the statement in the BEGINGRAPH block, but do not embed it in another GTL statement block. The ENTRYFOOTNOTE statement must be placed directly inside the BEGINGRAPH block. The following conditions apply to ENTRYFOOTNOTE statements.**

- All ENTRYFOOTNOTE statements that appear in the template are gathered and placed in the ENTRYFOOTNOTE area.
- The placement of an ENTRYFOOTNOTE statement is relevant only in relation to other ENTRYFOOTNOTE statements.
- As the number of ENTRYFOOTNOTE statements increases, the size of the graphical area is reduced.

Footnotes always span the entire width of the output. By default, footnotes are "center-aligned," based on the type of the outermost layout. The meaning of "center-aligned" varies by layout type and the number of columns in the layout:
<table>
<thead>
<tr>
<th>Layout Type</th>
<th>Default horizontal centering of footnotes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDDED</td>
<td>Centered on width of entire graph</td>
</tr>
<tr>
<td>OVERLAY3D</td>
<td></td>
</tr>
<tr>
<td>LATTICE (COLUMNS &gt; 1)</td>
<td></td>
</tr>
<tr>
<td>DATAPANEL (COLUMNS &gt; 1)</td>
<td></td>
</tr>
<tr>
<td>DATALATTICE (COLUMNS &gt; 1)</td>
<td></td>
</tr>
<tr>
<td>OVERLAY</td>
<td>Centered on the plot area</td>
</tr>
<tr>
<td>OVERLAYEQUATED</td>
<td></td>
</tr>
<tr>
<td>LATTICE (COLUMNS=1)</td>
<td></td>
</tr>
<tr>
<td>DATAPANEL (COLUMNS=1)</td>
<td></td>
</tr>
<tr>
<td>DATALATTICE (COLUMNS=1)</td>
<td></td>
</tr>
</tbody>
</table>

Example: ENTRYFOOTNOTE Statement

The following graph was generated by the “Example Program” on page 1290:

![Graph example]

Example Program

```plaintext
proc template;
  define statgraph entryfootnote;
  begingraph;
```

First entryfootnote statement
Second entryfootnote statement
Third entryfootnote statement
entryfootnote "First entryfootnote statement" ;

layout overlay;
   histogram weight;
endlayout;

entryfootnote "Second entryfootnote statement" ;
entryfootnote "Third entryfootnote statement" ;
endgraph;
end;
run;

proc sgrender data=sashelp.cars template=entryfootnote;
run;

ENTRYTITLE Statement
Displays a title.

Requirement: You must place the ENTRYTITLE statement directly inside the BEGINGRAPH block. Do not embed it in any other GTL statement block.

Syntax
ENTRYTITLE text-item <text-item ...> </options> ;

Summary of Optional Arguments

Appearance options
  BACKGROUNDCOLOR=style-reference | color
  specifies the color of the text background.
  BORDER=TRUE | FALSE
  specifies whether a border is displayed around the text.
  BORDERATTRS=style-element | style-element (line-options) | (line-options)
  specifies the attributes of the border line around the text.
  OPAQUE=TRUE | FALSE
  specifies whether the text background is opaque (TRUE) or transparent (FALSE).
  OUTERPAD=AUTO | dimension | (pad-options)
  specifies the amount of extra space to add outside the entry border.
  PAD=dimension | (pad-options)
  specifies the amount of extra space that is reserved inside the entry border.
  TEXTATTRS=style-element | style-element (text-options) | (text-options)
  as a statement option, specifies the properties of the text. As a prefix-option, specifies the properties of individual text-items.

Text options
  HALIGN=CENTER | LEFT | RIGHT
as a statement option, specifies the horizontal alignment of the text. As a prefix-option, specifies the horizontal alignment of individual text-items.

\textbf{HALIGN=}\texttt{CENTER} | \texttt{GRAPH}

specifies whether the text is centered automatically by the system or is always centered in the graph area.

\textbf{SHORTTEXT=}(\texttt{text-item <…text-item>})

specifies alternate text to use if the specified text is too long for the output width. If the shortened text is itself too long, then it is truncated.

\textbf{TEXTFITPOLICY=}\texttt{WRAP} | \texttt{SHORT} | \texttt{TRUNCATE}

specifies how to handle text that is too long to fit in the output width.

\textbf{Required Argument}

\texttt{text-item <text-item ...>}

specifies one or more pieces of text for the entry. Each \texttt{text-item} has the following form:

\tt\begin{verbatim}
<prefix-option <...prefix-option>>"string" | dynamic | character-expression | \{text-command}
\end{verbatim}

Each piece of text can have multiple prefix options that precede the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All \textit{text-items} with the same HALIGN= setting are concatenated into one string. Up to three strings can result for each ENTRY statement. Leading and trailing blanks in the concatenated string are always used.

When a prefix option is used, it applies not only to the text that immediately follows the prefix option, but also to all subsequent text strings and text commands. If the same prefix option appears more than once, then that prefix overrides the last used prefix of the same name for the subsequent text strings and text commands.

\textbf{Restriction}

The maximum length for the title text is 512 characters in SAS 9.4 and earlier releases. Starting with \texttt{SAS 9.4M1}, this restriction is removed.

\textbf{Requirements}

\textit{string} must be enclosed in quotation marks.

\textit{character-expression} must be enclosed in an EVAL function.

\textit{text-command} must be enclosed in braces.

\textbf{Note}

Leading spaces are preserved, and trailing spaces are stripped.

\textbf{See}

Chapter 10, “Managing Text Items,” on page 1265 for more information and several examples.

\textbf{Optional Arguments}

\textbf{BACKGROUND\texttt{COLOR=}style-reference | color}

specifies the color of the text background.

\textit{style-reference}

specifies a style reference in the form \texttt{style-element:style-attribute}.

Only the style-attribute named COLOR or CONTRASTCOLOR is used.

\textbf{Default}

The GraphBackground:Color style reference.
INTERACTION
	
**OPAQUE=** TRUE must be in effect in order for the color to be seen. By default, OPAQUE=FALSE.

**BORDER=** TRUE | FALSE

specifies whether a border is displayed around the text.

**Default** FALSE

**See** "boolean " on page 1479 for other Boolean values that you can use.

**BORDERATTRS=** style-element | style-element (line-options) | (line-options)

specifies the attributes of the border line around the text.

**Default** The GraphBorderLines style element.

**Interaction** BORDER= TRUE must be set for this option to have any effect.

**See** "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

"Line Options" on page 1522 for available line-options.

**HALIGN=** CENTER | LEFT | RIGHT

as a statement option, specifies the horizontal alignment of the text. As a prefix-option, specifies the horizontal alignment of individual text-items. Each text-item has a horizontal alignment, and text-items with the same alignment are always grouped together.

**Default** CENTER

**HALIGNCENTER=** AUTO | GRAPH

specifies whether the text is centered automatically by the system or is always centered in the graph area.

**AUTO**

specifies that the system determines how the text is centered.

The text is centered as follows:

- For LAYOUT GRIDDED, LAYOUT OVERLAY3D, and LAYOUT REGION layouts, and for LAYOUT LATTICE, LAYOUT DATAPANEL, and LAYOUT DATALATTICE layouts that have more than one column, center the text in the graph area.

- For LAYOUT OVERLAY and LAYOUT OVERLAYEQUATED layouts, and for LAYOUT LATTICE, LAYOUT DATAPANEL, and LAYOUT DATALATTICE layouts that have only one column, center the text in the graph wall area. If the length of the text exceeds the width of the graph wall area, then center the text in the graph area instead.

**GRAPH**

specifies that the text is always centered in the graph area.

**Default** AUTO

**Interaction** The prefix option HALIGN= must specify CENTER for this option to have any effect.
**OPAQUE=TRUE | FALSE**
specifies whether the text background is opaque (TRUE) or transparent (FALSE).

<table>
<thead>
<tr>
<th>Default</th>
<th>FALSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>When this option is set to FALSE, the background color is not used.</td>
</tr>
<tr>
<td>See</td>
<td>“boolean” on page 1479 for other Boolean values that you can use.</td>
</tr>
</tbody>
</table>

**OUTERPAD=**
Specifies the amount of extra space to add outside the entry border.

**AUTO**
specifies that the default outside padding for this component is used.

**dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

**(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can include one or more of the following pad options:

- **LEFT=**
specifies the amount of extra space to add to the left side.

- **RIGHT=**
specifies the amount of extra space to add to the right side.

- **TOP=**
specifies the amount of extra space to add to the top.

- **BOTTOM=**
specifies the amount of extra space to add to the bottom.

<table>
<thead>
<tr>
<th>Note</th>
<th>Sides that are not assigned padding are padded with the default amount.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>Use pad-options to create non-uniform padding.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default</th>
<th>AUTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note</td>
<td>The default units for dimension are pixels.</td>
</tr>
<tr>
<td>See</td>
<td>“dimension” on page 1480</td>
</tr>
</tbody>
</table>

**PAD=**
specifies the amount of extra space that is reserved inside the entry border.

**dimension**
specifies a dimension to use for the extra space at the left, right, top, and bottom of the entry border.

**(pad-options)**
a space-separated list of one or more name-value pair pad options, enclosed in parentheses.

The list can include one or more of the following pad options:
LEFT=\textit{dimension}  
specifies the amount of extra space to add to the left side.

RIGHT=\textit{dimension}  
specifies the amount of extra space to add to the right side.

TOP=\textit{dimension}  
specifies the amount of extra space to add to the top.

BOTTOM=\textit{dimension}  
specifies the amount of extra space to add to the bottom.

\textbf{Note}  Sides that are not assigned padding are padded with the default amount.

\textbf{Tip}  Use \textit{pad-options} to create non-uniform padding.

\textbf{Default}  LEFT=3, RIGHT=3, TOP=0, and BOTTOM=0

\textbf{Note}  The default units for \textit{dimension} are pixels.

\textbf{See}  "\textit{dimension} on page 1480"

\textbf{SHORTTEXT=}(\textit{text-item} <…\textit{text-item}>)

specifies alternate text to use if the specified text is too long for the output width. If the shortened text is itself too long, then it is truncated.

\textbf{Interactions}  This option has no effect unless \textbf{TEXTFITPOLICY= SHORT}.

\hspace{1cm} This option is ignored if any \textit{text-items} include an \textit{HALIGN=} prefix option.

\hspace{1cm} The horizontal alignment of the shortened text is derived from the horizontal alignment of the title to be shortened.

\textbf{TEXTATTRS=\textit{style-element} | \textit{style-element (text-options)} | (text-options)}
as a \textit{statement option}, specifies the properties of the text. As a \textit{prefix-option}, specifies the properties of individual \textit{text-items}.

\textbf{Default}  The GraphTitleText style element.

\textbf{Notes}  This option can be used as both a prefix option and a statement option. When used as a prefix option, it overrides the statement option.

\hspace{1cm} When used as a prefix option, TEXTATTRS= cancels the last used TEXTATTRS= prefix option. It resets all text options to those set by 1) the TEXTATTRS= statement option or 2) the default style element for the statement (GraphValueText) if no TEXTATTRS= statement option is used.

\textbf{See}  “General Syntax for Attribute Options” on page 1519 for the syntax on using a \textit{style-element}.

\hspace{1cm} “Text Options” on page 1525 for available \textit{text-options}.

\textbf{TEXTFITPOLICY=WRAP | SHORT | TRUNCATE}
specifies how to handle text that is too long to fit in the output width.

\textbf{WRAP}  
specifies that the text wraps to the next line(s).
Restriction  Text wrapping is available only for ENTRYTITLE statements that appear in the outermost layout.

**SHORT**
specifies that the text indicated by the SHORTTEXT= option be substituted when the text does not fit.

**Requirement**  You must specify the SHORTTEXT= option for this option to take effect.

**Note**  If the short text is also too long, then it is truncated.

**See**  “SHORTTEXT=(text-item <...text-item>)” on page 1287

**TRUNCATE**
specifies that the text is truncated to make it fit.

**Default**  WRAP

**Prefix Options**

**HALIGN=** CENTER | LEFT | RIGHT
specifies the horizontal alignment of the text that follows.

**See**  HALIGN= on page 1293

**TEXTATTRS=** style-element | style-element (text-options) | (text-options)
specifies the properties of the text that follows.

**See**  TEXTATTRS= on page 1295

**Text Commands**

**{ SUB "string" | dynamic }**
text-command that specifies that the string or dynamic is to appear as subscript text.

**See**  “Rules for Unicode and Special Character Specifications” on page 1270

**Example**  entrytitle "y = " b{sub "0"} " + b" {sub "1"} "x";

**{ SUP "string" | dynamic }**
text-command that specifies that the string or dynamic is to appear as superscript text.

**See**  “Rules for Unicode and Special Character Specifications” on page 1270

**Example**  entrytitle "R" {sup "2"} " = " RSQUARED;

**{ UNICODE "hex-string" | keyword | dynamic }**
text-command that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.
"hex-string"x
  a four-byte hexadecimal constant that represents a UNICODE character in
  the current font.

  For a complete listing, see http://unicode.org/charts/charindex.html.

keyword
  a SAS keyword for a UNICODE character.

  For a listing of keywords supplied by SAS, see “Reserved Keywords and
  Unicode Values” on page 1271.

dynamic
  a dynamic variable that resolves to either "hex-string"x or a keyword for a
  UNICODE character.

Note
  The UNICODE text command attempts to access a UNICODE value in
  the current font. Not all fonts support accessing characters through
  their UNICODE value. Some fonts support only a limited set of
  UNICODE values. If the UNICODE value is not accessible, then the
  command might be ignored or a nonprintable character might be
  substituted.

See
  “Using UNICODE Text” on page 1269

  “Rules for Unicode and Special Character Specifications” on page
  1270

Example
  entrytitle {unicode alpha} "=" CONF;
  entrytitle {unicode "03B1"x} "=" CONF;

Details

The ENTRYTITLE statement places title text above the graphical area. More than
one ENTRYTITLE statement can be used. Titles appear in the order of the
ENTRYTITLE statements within the template.

When you add an ENTRYTITLE statement to a template definition, place the
statement in the BEGINGRAPH block, but do not embed it in another GTL
statement block. The ENTRYTITLE statement must be placed directly inside the
BEGINGRAPH block. The following conditions apply to ENTRYTITLE statements.

- All ENTRYTITLE statements that appear in the template are gathered and
  placed in the ENTRYTITLE area.

- The placement of an ENTRYTITLE statement is relevant only in relation to other
  ENTRYTITLE statements.

- As the number of ENTRYTITLE statements increases, the size of the graphical
  area is reduced.
Titles always span the entire width of the output. By default, titles are "center-aligned," based on the type of the outermost layout. The meaning of "center-aligned" varies by layout type and the number of columns in the layout:

<table>
<thead>
<tr>
<th>Layout Type</th>
<th>Default horizontal centering of titles</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRIDDED</td>
<td>Centered on width of entire graph</td>
</tr>
<tr>
<td>OVERLAY3D</td>
<td></td>
</tr>
<tr>
<td>LATTICE (COLUMNS &gt; 1)</td>
<td></td>
</tr>
<tr>
<td>DATAPANEL (COLUMNS &gt; 1)</td>
<td></td>
</tr>
<tr>
<td>DATALATTICE (COLUMNS &gt; 1)</td>
<td></td>
</tr>
<tr>
<td>OVERLAY</td>
<td>Centered on the plot area</td>
</tr>
<tr>
<td>OVERLAYEQUATED</td>
<td></td>
</tr>
<tr>
<td>LATTICE (COLUMNS=1)</td>
<td></td>
</tr>
<tr>
<td>DATAPANEL (COLUMNS=1)</td>
<td></td>
</tr>
<tr>
<td>DATALATTICE (COLUMNS = 1)</td>
<td></td>
</tr>
</tbody>
</table>

Example: ENTRYTITLE Statement

The following graph was generated by the "Example Program" on page 1298:

Example Program

```r
proc template;
```
define statgraph entrytitle;
  begigraph;
    entrytitle "First entrytitle statement" ;
    layout overlay;
      histogram weight;
    endlayout;
    entrytitle "Second entrytitle statement" ;
    entrytitle "Third entrytitle statement" ;
  endgraph;
end;
run;

proc sgrender data=sashelp.cars template=entrytitle;
run;
Chapter 12

Custom Marker Definition Statements
Custom Marker Definition Statements

Dictionary

SYMBOLCHAR Statement

Defines a marker symbol using a Unicode character so that the symbol can be referenced in other statements.

**Requirement:**
The SYMBOLCHAR statement must be a child of the BEGINGRAPH block. It cannot be embedded in any other GTL statement block.

**Note:**
This statement is valid starting in SAS 9.4M1.

**Syntax**

```
SYMBOLCHAR NAME=marker-name CHAR=""hex-string | keyword | dynamic
<\ options>;
```

**Required Arguments**

**NAME=marker-name**
- specifies a name for the marker symbol. The name can be used in statements that support marker symbols.

**Interaction**
- If the name matches one of the system-provided symbol names such as CIRCLE, then the system symbol is replaced by the user-defined symbol. See “Marker Options” on page 1523 for a list of the system-provided symbols.

**Note**
- Do not enclose the name in quotation marks.

**See**
- “Details” on page 1305
The CHAR statement specifies a glyph (character) to be used as the marker symbol. You specify the character by using its Unicode specification or its keyword equivalent.

**CHAR="hex-string"x**

specifies a four-byte hexadecimal constant that represents a Unicode character in the current font.

You can find a complete listing of the Unicode hexadecimal constants at the following URL: http://www.unicode.org/charts/charindex.html

**keyword**

specifies a SAS keyword for a Unicode character.

**dynamic**

specifies a reference to a dynamic variable that resolves to either a "hex-string"x constant or a Unicode character keyword.

Tip

This statement attempts to access the specified Unicode value in the current font. Some fonts do not support accessing characters by using their Unicode value. Other fonts support only a limited set of Unicode values. If the Unicode value is not accessible, then this statement might be ignored or a nonprintable character might be substituted.

### Optional Arguments

**HOFFSET=number**

specifies a horizontal offset for the marker symbol.

<table>
<thead>
<tr>
<th>Default</th>
<th>0. The marker symbol is centered on its data point.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>-0.5–+0.5, where 0.5 represents one-half of the original marker size.</td>
</tr>
<tr>
<td>Interaction</td>
<td>Starting with SAS 9.4M3, the specified offset is also applied to the marker symbol that is displayed in the legend.</td>
</tr>
<tr>
<td>Notes</td>
<td>Prior to SAS 9.4M3, a positive offset moves the marker symbol bounding box to the right. A negative offset moves it to the left. Starting with SAS 9.4M3, a positive offset moves the marker symbol to the right within its bounding box, and a negative offset moves it to the left. The bounding box remains centered on the data point. After the offset, size, and rotation are applied to the marker symbol, any portion of the symbol that falls outside of the marker bounding box is clipped.</td>
</tr>
<tr>
<td>Tip</td>
<td>If clipping occurs, use this option, the VOFFSET= and SCALE= options, and the suboption SIZE= in the MARKERATTRS= option to remove the clipping.</td>
</tr>
</tbody>
</table>

**ROTATE=number**

specifies the angle of rotation for the marker symbol in degrees. Positive angles are measured in the counter-clockwise direction, and negative angles are measured in the clockwise direction.

| Default       | 0. No rotation is performed. |
Note An angle that exceeds 360 degrees in absolute value can be specified.

**SCALE=number**
specifies a scale factor for the marker symbol as a percentage. The scale factor is applied to the character's height and width.

Default 1.0 (100%)

Note The outer edges of the image might be clipped by the bounding box when a large scale factor is specified.

**TEXTATTRS=style-element | style-element (text-options) | (text-options)**
specifies the text attributes for the character symbol.

Default The GraphUnicodeText style element.

Restriction Only the text attributes FAMILY=, STYLE=, and WEIGHT= are used. The COLOR= and SIZE= text attributes are derived from the plot statement's MARKERATTRS= option.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text-options.

**VOFFSET=number**
specifies a vertical offset for the marker symbol.

Default 0. The marker symbol is centered on its data point.

Range -0.5–+0.5, where 0.5 represents one-half of the original marker size.

Interaction Starting with SAS 9.4M3, the specified offset is also applied to the marker symbol that is displayed in the legend.

Notes Prior to SAS 9.4M3, a positive offset moves the marker symbol bounding box up. A negative offset moves it down.

Starting with SAS 9.4M3, a positive offset moves the marker symbol up within its bounding box, and a negative offset moves it down. The bounding box remains centered on the data point. After the offset, size, and rotation are applied to the marker symbol, any portion of the symbol that falls outside of the marker bounding box is clipped.

Tip If clipping occurs, use this option, the HOFFSET= and SCALE= options, and the suboption SIZE= in the MARKERATTRS= option to remove the clipping.

**Details**

The SYMBOLCHAR statement defines a custom marker symbol from a Unicode character. You can use the marker symbol that is created in the following options:

- the DATASYMBOLS= option in the BEGINGRAPH statement
- the SYMBOL= suboption of the MARKERATTRS= option, which is supported by the following statements:
Symbol specifications are not checked for uniqueness. More than one SYMBOLCHAR statement can define the same character. Therefore, you can use SYMBOLCHAR statements in IF/ELSE statements. Symbol specifications also are not validated at compile time. An invalid specification might not generate a warning when the output is rendered and might create unexpected results.

You can use the COLOR=, SIZE=, and TRANSPARENCY= suboptions of the MARKERATTRS= option to modify the appearance of markers that are created by the SYMBOLCHAR statement. The WEIGHT= suboption has no effect on these markers.

The markers are clipped to the original marker size after rotation, scaling, and offsets are applied. If clipping occurs, then you can use the SIZE= suboptions of the MARKERATTRS= option in the plot statement and the SCALE= option in the SYMBOLCHAR statement to adjust the size and scaling to eliminate the clipping.

Example: SYMBOLCHAR Statement

This example shows how to create marker symbols from the Unicode ballot X, heavy character (‘2718x’) and check mark, heavy character (‘2718x’). It also shows you how to use the symbols in a scatter plot and how to add them to a legend. In this example, a grouped bar chart shows the daily failure count for two processes, Process A and Process B. A scatter plot is overlaid on the bar chart in order to show the total failure count above each bar. Acceptable counts are indicated by a green Unicode check mark. High counts are indicated by a red Unicode ballot X. Finally, a legend is used to describe the custom symbols. The following figure shows the output.

A failure count that is more than 120% of the target is considered high.
Example Program

/* Summarize the data in SASHELP.FAILURE to compute the failure count by day and process */
proc sort data=sashelp.failure out=temp;
   by day process;
run;

proc summary data=temp;
   by day process;
   var count;
   output out=failure sum=count;
run;

/* Determine the daily status for each process:
count at or under 120% of target is NORM, over is HIGH */
%let target=25;
data failure;
   length stat $20;
   set failure;
   if (count <= (&target*1.2)) then
      stat=strip(cat(substr(process,index(process,' '),2),"-NORM");
   if (count > (&target*1.2)) then
      stat=strip(cat(substr(process,index(process,' '),2),"-HIGH");
   label day="Weekday" count="Failure Count" stat="Status";
run;

/* Define the template */
proc template;
   define statgraph failures;
   begingraph;

   /* Define the norm/high marker symbols */
   symbolchar
      name=norm char='2714'x;
   symbolchar
      name=high char='2718'x;

   /* Create legend entries for the norm/high markers */
   legenditem type=marker name="norm" / markerattrs=(symbol=norm size=15pt color=green)
      label="Normal Failure Count";
   legenditem type=marker name="high" / markerattrs=(symbol=high size=15pt color=red)
      label="High Failure Count";

   /* Define an attribute map for the status markers */
   discreteattrmap name="statsymmap";
      value "A-NORM" / markerattrs=(symbol=norm color=green);
      value "B-NORM" / markerattrs=(symbol=norm color=green);
      value "A-HIGH" / markerattrs=(symbol=high color=red);
      value "B-HIGH" / markerattrs=(symbol=high color=red);
   enddiscreteattrmap;
   discreteattrvar attrvar=status var=stat attrmap="statsymmap";

   /* Define the graph */
   entrytitle "Daily Failure Report By Process";
   entryfootnote "A failure count that is more than 120% of the target is considered high.";
   endgraph;
end;
Program Description

Here is the SAS code for this example.

Create the graph data. The data in Sashelp.Failure is the source data. To get the daily counts, the data is summarized by weekday and process. The Stat column is then added to indicate whether the count is acceptable or high. Acceptable values are values that do not exceed 120% of the target value. Values over 120% are considered high. Acceptable values are A-NORM and B-NORM, and high values are A-HIGH and B-HIGH, where A and B identify the process.

/* Summarize the data in SASHELP.FAILURE to compute the failure count by day and process */
proc sort data=sashelp.failure out=temp;
   by day process;
run;
proc summary data=temp;
   by day process;
   var count;
   output out=failure sum=count;
run;

/* Determine the daily status for each process: count at or under 120% of target is NORM, over is HIGH */
%let target=25;
data failure;
length stat $20;
set failure;
if (count <= (&target*1.2)) then
   stat=strip(cat(substr(process,index(process,' '),2),"-NORM"));
if (count > (&target*1.2)) then
   stat=strip(cat(substr(process,index(process,' '),2),"-HIGH"));
label day="Weekday" count="Failure Count" stat="Status";
run;

Open the template definition.

/* Define the template */
proc template;
define statgraph failures;
Define the custom marker symbols. The NAME= option specifies an identifier for the markers, and the CHAR= option specifies the Unicode value. Unicode value ‘2714’x specifies a heavy check mark, and value ‘2718’x specifies a heavy ballot X.

/* Define the norm/high marker symbols */
symbolchar name=norm char='2714'x;
symbolchar name=high char='2718'x;

Create legend entries for the custom markers. Suboption SYMBOL= in the MARKERATTRS= option specifies the identifier of the custom markers.

/* Create legend entries for the norm/high markers */
legenditem type=marker name="norm" /
   markerattrs=(symbol=norm size=15pt color=green) label="Normal Failure Count";
legenditem type=marker name="high" /
   markerattrs=(symbol=high size=15pt color=red) label="High Failure Count";

Create a discrete attribute map to map the custom markers to the Stat column values. Suboption SYMBOL= in the MARKERATTRS= option specifies the identifier of the custom markers. Suboption COLOR= sets the color of the Unicode marker character and the associated data label. The DISCRETEATTRVAR statement creates attribute map variable STATUS.

/* Define an attribute map for the status markers */
discreteattrmap name="statsymmap";
   value "A-NORM" / markerattrs=(symbol=norm color=green);
   value "B-NORM" / markerattrs=(symbol=norm color=green);
   value "A-HIGH" / markerattrs=(symbol=high color=red);
   value "B-HIGH" / markerattrs=(symbol=high color=red);
enddiscreteattrmap;
discreteattrvar attrvar=status var=stat attrmap="statsymmap";

Define the graph and close the template definition. The SCATTERPLOT statement GROUP= option references the STATUS attribute map variable. The EVAL statement in the Y= option positions each character just above the top of its bar. The SIZE= suboption of the MARKERATTRS= option sets the marker size. The custom markers are included in the DISCRETELEGEND statement.

/* Define the graph */
entrytitle "Daily Failure Report By Process";
entryfootnote "A failure count that is more than 120% of the target is considered high.";
layout overlay / yaxisopts=(offsetmax=0.1);
barchartparm category=day response=count / name="barchart"
   group=process groupdisplay=cluster;
scatterplot x=day y=eval(count+2) / datalabel=count group=status
   groupdisplay=cluster markerattrs=(size=15pt);
   referenceline y=&target /
      lineattrs=(color=lightred pattern=dot) curvelabel="Target";
discretelegend "barchart" "norm" "high" / down=2 order=columnmajor;
endlayout;
discretelegend;
endgraph;
run;
Generate the graph.

```sas
proc sgrender data=failure template=failures;
run;
```

**SYMBOLIMAGE Statement**

Defines a marker symbol using an image file so that the image can be referenced in other statements.

**Requirement:**
The SYMBOLIMAGE statement must be a child of the BEGINGRAPH block. It cannot be embedded in any other GTL statement block.

**Note:**
This statement is valid starting in **SAS 9.4M1**.

**Syntax**

```sas
SYMBOLIMAGE
   NAME=marker-name
   IMAGE="image-file-spec" </options>;
```

**Required Arguments**

**NAME=** *marker-name*

specifies a name for the marker symbol. The name can be used in statements that support marker symbols.

**IMAGE=** "image-file-spec"

specifies the name and location of the image file. The supported image types are GIF, JPEG, and PNG.

**Interaction**

If the name matches one of the system-provided symbol names such as CIRCLE, then the system symbol is replaced by the user-defined symbol. See "Marker Options" on page 1523 for a list of the system-provided symbols.

**Note**

Do not enclose the name in quotation marks.

**See**

"Details" on page 1311

**Optional Arguments**

**HOFFSET=** *number*

specifies a horizontal offset for the marker symbol.

**Default**

0. The marker symbol is centered on its data point.

**Range**

-0.5→+0.5, where 0.5 represents one-half of the original marker size.
Notes A positive offset moves the marker symbol bounding box to the right. A negative offset moves it to the left.

The specified offset is not applied to the image symbol that is displayed in the legend.

**ROTATE**=*number*
specifies the angle of rotation for the marker symbol in degrees. Positive angles are measured in the counter-clockwise direction, and negative angles are measured in the clockwise direction.

Default 0. No rotation is performed.

Note An angle that exceeds 360 degrees in absolute value can be specified.

**SCALE**=*number*
specifies a scale factor for the marker symbol as a percentage. The scale factor is applied to the character's height and width.

Default 1.0 (100%)

Note The outer edges of the image might be clipped by the bounding box when a large scale factor is specified.

**VOFFSET**=*number*
specifies a vertical offset for the marker symbol.

Default 0. The marker symbol is centered on its data point.

Range -0.5→0.5, where 0.5 represents one-half of the original marker size.

Notes A positive offset moves the marker symbol bounding box up. A negative offset moves it down.

The specified offset is not applied to the image symbol that is displayed in the legend.

Details

The SYMBOLIMAGE statement defines a custom marker symbol from an image that is stored in an image file. The image file must exist on the local file system. URL access is not supported. The supported image formats are GIF, JPG, and PNG. The marker symbol that is created can be used in the following options:

- the DATASYMBOLS= option in the BEGINGRAPH statement
- the SYMBOL= suboption of the MARKERATTRS= option, which is supported by the following statements:
  - DISCRETELEGEND (VALUE statement)
  - LEGENDITEM
  - SCATTERPLOT
  - SCATTERPLOTMATRIX
  - LINECHART
  - SERIESPLOT
  - NEEDLEPLOT
  - STEPPLOT

Symbol specifications are not checked for uniqueness. More than one SYMBOLIMAGE statement can define the same character. Therefore, you can use SYMBOLIMAGE statements in IF/ELSE statements.
The following options normally affect the appearance of markers. However, they have no effect on image marker symbols that are created by the SYMBOLIMAGE statement.

- the BEINGRAPH statement DATACONTRASTCOLORS= option
- the COLOR= and WEIGHT= suboptions of the MARKERATTRS= option
- the FILLEDOUTLINEDMARKERS= option

The SIZE= and TRANSPARENCY= suboptions of the MARKERATTRS= option do affect the appearance of markers that are created by the SYMBOLIMAGE statement.

The markers are clipped to the original marker size after rotation, scaling, and offsets are applied. If clipping occurs, then you can use the SIZE= suboptions of the MARKERATTRS= option in the plot statement and the SCALE= option in the SYMBOLIMAGE statement to adjust the size and scaling to eliminate the clipping.

Example: SYMBOLIMAGE Statement

This example shows how to create marker symbols from GIF icon image files. It also shows you how to use the symbols in a scatter plot and how to add them to a legend. In this example, a grouped bar chart shows the daily failure count for two processes, Process A and Process B. A scatter plot is overlaid on the bar chart to show the total failure count above each bar. Acceptable counts are indicated by a green check mark icon. High counts are indicated by a caution icon. Finally, a legend is used to describe the custom symbols. The following figure shows the output.

Example Program

/* Summarize the data in SASHELP.FAILURE to compute the
failure count by day and process */
proc sort data=sashelp.failure out=temp;
  by day process;
run;

proc summary data=temp;
  by day process;
  var count;
  output out=failure sum=count;
run;

/* Determine the daily status for each process:
count at or under 120% of target is NORM, over is HIGH */
%let target=25;
data failure;
  length stat $20;
  set failure;
  if (count <= (&target*1.2)) then
    stat=strip(cat(substr(process,index(process,' '),2),"-NORM"));
  if (count > (&target*1.2)) then
    stat=strip(cat(substr(process,index(process,' '),2),"-HIGH"));
  label day="Weekday" count="Failure Count" stat="Status";
run;

/* Define the template */
proc template;
define statgraph failures;
begingraph;
  /* Define the norm/high marker symbols */
symbolimage name=norm image="C:\temp\check_green.gif" /
    voffset=0.5;
symbolimage name=high image="C:\temp\alert_orange.gif" /
    voffset=0.5;
  /* Create legend entries for the norm/high markers */
legenditem type=marker name="norm" /
  markerattrs=(symbol=norm size=16px)
  label="Normal Failure Count";
legenditem type=marker name="high" /
  markerattrs=(symbol=high size=16px)
  label="High Failure Count";
  /* Define an attribute map for the status markers
  Note: COLOR= affects marker labels only in this case. */
discreteattrmap name="statsymmap";
  value "A-NORM" / markerattrs=(symbol=norm color=green);
  value "B-NORM" / markerattrs=(symbol=norm color=green);
  value "A-HIGH" / markerattrs=(symbol=high color=orange);
  value "B-HIGH" / markerattrs=(symbol=high color=orange);
enddiscreteattrmap;
  discreteattrvar attrvar=status var=stat attrmap="statsymmap";
  /* Define the graph */
entrytitle "Daily Failure Report By Process"
  entryfootnote "A failure count that is more than 120% of the target is considered high."
layout overlay / yaxisopts=(offsetmax=0.1);
  barchartparm category=day response=count / name="barchart"
Program Description

Here is the SAS code for this example.

Create the graph data. The data in Sashelp.Failure is the source data. To get the daily counts, the data is summarized by weekday and process. The Stat column is then added to indicate whether the count is acceptable or high. Acceptable values are values that do not exceed 120% of the target value. Values over 120% are considered high. Acceptable values are A-NORM and B-NORM, and high values are A-HIGH and B-HIGH, where A and B identify the process.

/* Summarize the data in SASHELP.FAILURE to compute the failure count by day and process */
proc sort data=sashelp.failure out=temp;
  by day process;
run;

proc summary data=temp;
  by day process;
  var count;
  output out=failure sum=count;
run;

/* Determine the daily status for each process: count at or under 120% of target is NORM, over is HIGH */
%let target=25;
data failure;
  length stat $20;
  set failure;
  if (count <= (&target*1.2)) then
    stat=strip(cat(substr(process,index(process,' '),2),"-NORM"));
  if (count > (&target*1.2)) then
    stat=strip(cat(substr(process,index(process,' '),2),"-HIGH"));
  label day="Weekday" count="Failure Count" stat="Status";
run;

Open the template definition.

/* Define the template */
proc template;
  define statgraph failures;
  begingraph;
Define the custom marker symbols. The NAME= option specifies an identifier for the markers, and the IMAGE= option specifies the image file. The VOFFSET= option raises the markers 50% from their data points in order to prevent them from overlapping the bars in the bar chart.

```plaintext
/* Define the norm/high marker symbols */
symbolimage name=norm image="C:\temp\check_green.gif" / voffset=0.5;
symbolimage name=high image="C:\temp\alert_orange.gif" / voffset=0.5;
```

Create legend entries for the custom markers. Suboption SYMBOL= in the MARKERATTRS= option specifies the identifier of the custom markers.

```plaintext
/* Create legend entries for the norm/high markers */
legenditem type=marker name="norm" / markerattrs=(symbol=norm size=16px) label="Normal Failure Count";
legenditem type=marker name="high" / markerattrs=(symbol=high size=16px) label="High Failure Count";
```

Create a discrete attribute map to map the custom markers to the Stat column values. Suboption SYMBOL= in the MARKERATTRS= option specifies the identifier of the custom markers. Suboption COLOR= sets the color of the associated data label. It does not affect the color of the image. The DISCRETEATTRVAR statement creates attribute map variable STATUS.

```plaintext
/* Define an attribute map for the status markers
Note: COLOR= affects marker labels only in this case. */
discreteattrmap name="statsymmap";
   value "A-NORM" / markerattrs=(symbol=norm color=green);
   value "B-NORM" / markerattrs=(symbol=norm color=green);
   value "A-HIGH" / markerattrs=(symbol=high color=orange);
   value "B-HIGH" / markerattrs=(symbol=high color=orange);
enddiscreteattrmap;
discreteattrvar attrvar=status var=stat attrmap="statsymmap";
```

Define the graph and close the template definition. The SCATTERPLOT statement GROUP= option references the STATUS attribute map variable. The SIZE= suboption of the MARKERATTRS= option sets the marker size. The custom markers are included in the DISCRETELEGEND statement.

```plaintext
/* Define the graph */
entrytitle "Daily Failure Report By Process";
entryfootnote "A failure count that is more than 120% of the target is considered high."
layout overlay / yaxisopts=(offsetmax=0.1);
   barchartparm category=day response=count / name="barchart"
      group=process groupdisplay=cluster;
   scatterplot x=day y=count / datalabel=count group=status
      groupdisplay=cluster markerattrs=(size=16px);
   referenceline y=&target / lineattrs=(color=lightred pattern=dot) curvelabel="Target";
   discretelegend "barchart" "norm" "high" / down=2
      order=columnmajor;
endlayout;
endgraph;
end;
run;
```
Generate the graph.

proc sgrender data=failure template=failures;
run;
Key Concepts for Using Draw Statements

Introduction

The draw statements enable you to customize a graph by drawing visual elements anywhere within the graph. The following sections provide a comprehensive example and a brief overview to the major drawing concepts. For more details about the individual draw statements and a simple example of each, see the discussion for each individual statement. For complete usage information for the draw statements, consult the SAS Graph Template Language: User’s Guide.

Example

The following example shows a bar chart of the top global automobile makers in 2008. To focus the graph on the 2008 merger between Chrysler and Fiat, the example uses DRAWRECTANGLE to highlight the three bars in the chart that correspond to the unit sales for the two auto makers. The example also uses a DRAWARROW statement and two BEGINPOLYLINE blocks to point to the sales figures, and it uses a DRAWTEXT statement to explain the implications of the merger.

All of the draw statements in this example use the data space (see “About the Drawing Space and Drawing Units” on page 1322) to integrate the drawn elements into the graph’s data area.
/* Create the data and the macro variables for Chrysler, Fiat, and Chrysler+Fiat widths */
data mydata;
    length automaker $30;
    input automaker $ 1-30 million_units;
    FORMAT million_units 3.1;
    select (automaker);
        when ("Fiat") do;
            colorvar=1;
            call symput( "fwidth",  million_units);
        end;
        when ("Chrysler") do;
            colorvar=1;
            call symput( "cwidth",  million_units);
        end;
        when ("Fiat + Chrysler") do;
            colorvar=2;
            call symput( "cfwidth",  million_units);
        end;
        otherwise colorvar=3;
    end;
datalines;
Toyota                         8.7
GM                             7.7
Volkswagen                     6.0
Renault-Nissan                 5.8
Ford                           5.4
Fiat + Chrysler                4.5
Hyundai                        4.2
Honda                          3.8
PSA                            3.2
Fiat                           2.5
Suzuki                         2.4
Chrysler                       2.0
Daimler                        1.9
BMW                            1.4
Mazda                          1.4
Mitsubishi                     1.1

/* Create template definition */
proc template;
define statgraph automerger;
begingraph / drawspace=datavalue;
  entrytitle halign=center 'Top Global Automakers (2008 Annual Unit Sales)';
  layout lattice / rowdatarange=data columndatarange=data rowgutter=10 columngutter=10;
  layout overlay / xaxisopts=(label=('Units (millions)'))
    yaxisopts=(reverse=true display=(ticks tickvalues line));
  barchart category=automaker response=million_units / group=colorvar
    name='bar(h)' barlabel=true datakin=pressed orient=horizontal;
drawrectangle x=eval(&cwidth/2.0) y="Chrysler"
  width=&cwidth height=0.95 / widthunit=data heightunit=data
  display=(outline) outlineattrs=(color=yellow);
drawrectangle x=eval(&fwidth/2.0) y="Fiat"
  width=&fwidth height=0.95 / widthunit=data heightunit=data
  display=(outline) outlineattrs=(color=yellow);
drawrectangle x=eval(&cfwidth/2.0) y="Fiat + Chrysler"
  width=&cfwidth height=0.95 / widthunit=data heightunit=data
  display=(outline) outlineattrs=(color=yellow);
beginpolyline x=eval(&cwidth + 0.5) y="Chrysler";
draw x=eval(&cwidth + 1.5) y="Chrysler";
draw x=eval(&cwidth + 1.5) y="Fiat";
draw x=eval(&fwidth + 0.5) y="Fiat";
endpolyline;
beginpolyline x=eval(&cwidth + 1.5) y="Suzuki";
draw x=eval(&cfwidth + 1.5) y="Suzuki";
draw x=eval(&cfwidth + 1.5) y="Fiat + Chrysler";
endpolyline;
drawarrow x1=eval(&cfwidth + 1.5) x2=eval(&cfwidth + 0.5)
  y1="Fiat + Chrysler" y2="Fiat + Chrysler"
  / arrowheadscale=0.5 arrowheadshape=barbed;
drawtext "Alliance creates the #6 Global Automaker by volume" /
  y="Honda" x=eval(&cfwidth+2.5) width=2 widthunit=data;
endlayout;
endgraph;
Types of Elements That Can Be Drawn

The GTL draw statements enable you to draw the following types of elements in a graph:

- text
- arrows and lines
- geometric shapes like ovals, rectangles, polygons, and polylines (a set of connecting lines)
- images

Using these elements individually and especially by combining them, you can describe the non-data aspects of your graph. For example, you can display a company logo in a specified location within the graph. Or you can create custom features that are difficult to create by other means—for example, you could draw a broken axis. Using the draw statements creatively, you can direct viewer attention to features of interest in the graph by drawing elements that highlight those features.

About the Drawing Space and Drawing Units

Each draw statement positions a drawn element using Cartesian coordinates that you specify as X and Y values in the statement. The coordinates that you specify are positioned relative to the drawing space that is in effect for the statement. The available drawing spaces are the data area, the wall area, the layout area, and the graph area. The coordinates can be specified in pixel units, percentage units, or as values that are in the units of the data (available only in the data area).

The images that accompany the following descriptions all use pixel data units. Each, however, was drawn in a different drawing space using this same LAYOUT OVERLAY definition block:

```plaintext
layout overlay /
  xaxisopts=(display=(line ticks tickvalues))
  yaxisopts=(display=(line ticks tickvalues));
scatterplot x=height y=weight;
drawrectangle x=0 y=0 width=50 height=50 /
  anchor=bottomleft display=(fill) fillattrs=(color=green)
  transparency=0.75 widthunit=pixel heightunit=pixel;
drawline x1=0 y1=0 x2=18 y2=18 / lineattrs=(color=red);
endlayout;
```
### Table 13.1 Drawing Spaces

<table>
<thead>
<tr>
<th>Drawing Space</th>
<th>Drawing Units</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Area</td>
<td>VALUE, PERCENT, PIXEL</td>
<td>The area where data is displayed in the graph, honoring the offsets that are set for the axes. For a discussion on axis offsets, see “Adjusting Axis Offsets” on page 985. The data area does not apply to graphs that do not have axes, such as pie charts, which must be drawn in a REGION layout. For graphs produced within LAYOUTDATAPANEL and LAYOUTDATALATTICE layouts, drawn elements are clipped if they extend outside of the wall boundaries.</td>
<td><img src="image" alt="Data Area" /></td>
</tr>
<tr>
<td>Wall Area</td>
<td>PERCENT, PIXEL</td>
<td>The area bounded by orthogonal axis pairs, ignoring the offsets that are set for the axes. In two-dimensional graphs, there is one wall bounded by the XY axes, including the secondary axes, if used. In three-dimensional graphs, there are three walls, bounded by the XY, YZ, and XZ axes. The wall area does not apply to graphs that do not have axes, such as pie charts, which must be drawn in a REGION layout. For graphs produced within LAYOUTDATAPANEL and LAYOUTDATALATTICE layouts, drawn elements that extend outside of the wall boundaries are clipped.</td>
<td><img src="image" alt="Wall Area" /></td>
</tr>
<tr>
<td>Layout Area</td>
<td>PERCENT, PIXEL</td>
<td>The entire area of the layout container that is the immediate parent container of the draw statement. The figure to the right shows the case where a LAYOUT OVERLAY is the draw statement’s layout container. Titles and footnotes are always displayed outside of the outermost layout, so those areas are never part of the layout drawing space.</td>
<td><img src="image" alt="Layout Area" /></td>
</tr>
<tr>
<td>Drawing Units</td>
<td>Drawing Space</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>PERCENT, PIXEL</td>
<td>Graph Area</td>
<td>The entire area that is available to the graph display, whether a single-cell or multi-cell graph. Because the graph drawing space spans the entire graph, the location of the drawn element in the graph is independent of the draw statement’s placement within the template definition, even if the draw statement is specified within a nested layout.</td>
<td></td>
</tr>
</tbody>
</table>

To specify the drawing space for both the X and the Y dimension, you use the DRAWSPACE= option. To specify the drawing space individually for either the X or the Y dimension, you use the options XSPACE=, X1SPACE=, X2SPACE=, YSPACE=, Y1SPACE=, or Y2SPACE=, depending on the draw statement. The value that you set on any of these options is a single composite value that specifies both the drawing space and the drawing units in the following format:

\[<\text{DrawingSpace}>\langle\text{Units}\rangle\]

For example, DRAWSPACE=GRAPHPIXEL specifies the GRAPH drawing space with PIXEL drawing units, indicating that the statement’s X,Y coordinates are expressed in pixels. Similarly, DRAWSPACE=LAYOUTPERCENT specifies the LAYOUT drawing space with PERCENT drawing units, indicating that the statement’s X,Y coordinates are expressed as percentages.

The global DRAWSPACE is LAYOUTPERCENT. The global DRAWSPACE= setting for all of the draw statements is LAYOUTPERCENT. The draw statements inherit the global setting from the DRAWSPACE= setting in the BEGINGRAPH statement.

- To change the global drawing space and drawing units for all of the draw statements within the template definition, use the DRAWSPACE= option in the BEGINGRAPH statement.
- To change the default for an individual draw statement, use that statement’s DRAWSPACE= option. If needed, you can specify different settings for the different draw statements within the template definition.

For the DATA drawing space, the VALUE drawing units specify that the coordinates are expressed as values along the axis. When you specify the DATA drawing space, you can use the draw statement’s XAXIS= and YAXIS= options to specify which axis scale to use for the coordinates.

- If the specified axis does not exist in the plot or is not valid for the draw statement's layout container, then the draw statement is ignored.
- For a discrete axis, if the statement's specified X or Y value does not exist in the data, then the draw statement is ignored.
- For a continuous axis, if the statement's specified X or Y value does not exist in the data, then the value is extrapolated.

When specifying the drawing space and drawing units, you can set a common setting for all of the X and Y coordinates. Or you can specify different settings for
each individual coordinate. The DRAWSPACE= setting in the BEGINGRAPH statement applies the global space and unit settings to all of the draw statements within the BEGINGRAPH/ENDGRAPH block. The DRAWSPACE= setting in an individual draw statement applies the space and unit setting only to the coordinate(s) for that statement. Thus, for lines and arrows, the setting applies to both the X1, Y1 coordinate and the X2, Y2 coordinate.

To specify the drawing space and drawing units separately for the X coordinate and for the Y coordinate, use the XSPACE=, YSPACE=, X1SPACE=, Y1SPACE=, X2SPACE=, and Y2SPACE= options, as applicable, in each draw statement. These options override the DRAWSPACE= option.

---

**How the Drawn Elements Are Anchored**

When you specify the X and Y coordinates in a draw statement, the element is drawn from an anchor point that is placed in the drawing area at the X,Y coordinate point.

- For lines and arrows, the anchor point is the starting point of the line or arrow, which is specified with the draw statement’s X1= and Y1= values.
- For elements that have height and width, the anchor point can be one of the following points:

```
TOPLEFT  TOP  TOPRIGHT
  LEFT  CENTER  RIGHT
BOTTOMLEFT  BOTTOM  BOTTOMRIGHT
```

The default anchor position is CENTER. To change the anchor position, use the draw statement’s ANCHOR= option.

---

**About Drawing Layers**

A draw statement can draw its element in either of two “layers” in the graph: the front layer or the back layer. By default, the statement draws in the front layer, which places the element in front of all other graphics elements, including data points, data labels, axis labels, and so on. In some cases, this might cause the drawn element to block the view of other graphics elements in the graph.

To prevent a drawn element for blocking the view of other graphics elements in the graph, you can use the draw statement’s TRANSPARENCY= option to add transparency to the drawn element. With an appropriate transparency setting, you should be able to see any graphics elements that are behind the drawn element.
Alternatively, you can use the draw statement’s LAYER= option to draw the element in the back layer, which places the drawn element behind all other graphics elements in the graph. For example, the following figure shows two different versions of a graph that uses DRAWLINE to draw a diagonal line across the axis area. The version to the left draws the line in the front layer, which causes the line to cover some of the data labels in the graph and portions of the series line. The version to the right uses LAYER=BACK to draw the line in the back layer. This prevents the line from covering the data labels and the portions of the series line.

The following code fragment shows the code that positions the line in the back layer:

```plaintext
layout overlay / walldisplay=(outline)
    xaxisopts=(griddisplay=on display=(line ticks tickvalues))
    yaxisopts=(griddisplay=on display=(line ticks tickvalues));
seriesplot x=open y=close / datalabel=date;
drawline x1=0 y1=0 x2=100 y2=100 /
    x1space=wallpercent y1space=wallpercent
    x2space=wallpercent y2space=wallpercent
    lineattrs=(color=cyan thickness=6) layer=back ;
endlayout;
```

**The Back Layer is Behind the Background.** Although drawing elements in the back layer prevents them from obstructing other data elements in the graph, it is not always the right solution to the problem.

If a draw statement uses LAYER=BACK, then it draws the element behind all other graphics elements, such as the layout background or a discrete legend’s background. To ensure that the element is visible in the graph, you might have to do one or more of the following:

- In overlay-type layouts or in a SCATTERPLOTMATRIX, you can use the parent layout’s WALLDISPLAY= option to turn off the display of the plot wall. In the example code just shown, WALLDISPLAY=(OUTLINE) displays an outline around the graph wall but does not display the wall fill. Suppressing the fill ensures that the drawn line is visible behind the plot wall.
If a layout container uses OPAQUE=TRUE so that it can set visual attributes for the background, then the opaque background covers and therefore hides any drawn element that is behind the background. When assigning visual attributes to a graph background, therefore, it might be better to use TRANSPARENCY= rather than LAYER= to prevent drawn elements from covering other graphics elements in the graph.

If the results of a plot statement or other GTL-statement covers the drawn element, then you can use transparency to reveal the drawn element. For example, you could use a plot statement’s DATATRANSPARENCY= option to set an appropriate transparency level for the plot.
Draw Statements

Dictionary

BEGINPOLYGON Statement
Defines the starting point for drawing a polygon in the graph.

Syntax
BEGINPOLYGON X=constant | scalar-expression
Y=constant | scalar-expression </options>;
DRAW X=constant | scalar-expression
Y=constant | scalar-expression </options>;
<… more DRAW statements …>
ENDPOLYGON;

Summary of Optional Arguments

Appearance options
DISPLAY=STANDARD | ALL | (display-options)
specifies the features to display for the polygon.
FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the fill attributes for this BEGINPOLYGON block.
LAYER=FRONT | BACK
specifies the layer on which this BEGINPOLYGON block’s output is drawn.

**OUTLINEATTRS=** *style-element | style-element (line-options) | (line-options)*

specifies the outline attributes for this BEGINPOLYGON block.

**TRANSPARENCY=** *number*

specifies the degree of the transparency of this BEGINPOLYGON block’s output.

**Axes options**

**DISCRETEOFFSET=** *number*

specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

**XAXIS=** *X | X2*

specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

**YAXIS=** *Y | Y2*

specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

**Drawing space options**

**DRAWSPACE=** *GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE*

specifies a default drawing space and drawing units for this BEGINPOLYGON block.

**XSPACE=** *GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE*

specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

**YSPACE=** *GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE*

specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

**ODS options**

**URL=** *"string" | string-expression*

specifies an HTML page that is displayed when the output of this draw statement is selected.

**Required Arguments**

The following options are required in the BEGINPOLYGON statement and the DRAW statement.

**X=** *constant | scalar-expression*

specifies the X value for a point in the polygon. When used in the BEGINPOLYGON statement, it specifies the X value of the starting point of the polygon.
**Interaction** The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

**Note** When XSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

\[ Y= \text{constant} \mid \text{scalar-expression} \]

Specifies the Y value for a point in the polygon. When used in the BEGINPOLYGON statement, it specifies the Y value of the starting point of the polygon.

**Interaction** The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

**Note** When YSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

### Optional Arguments

The following options can be used in the BEGINPOLYGON statement.

**DISCRETEOFFSET=number**

Specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

- **Default** 0 (no offset, output is centered on the discrete X values, or the discrete Y values, or both)
- **Range** –0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the right. If the Y axis is discrete, then a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.
- **Restriction** This option applies only when the options XSPACE= or YSPACE= use DATAVALUE, and when X or Y are values on a discrete axis. For nondiscrete axes, this option is ignored.
- **Tip** Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**DISPLAY=STANDARD | ALL | (display-options)**

Specifies the features to display for the polygon.

**STANDARD**

displays an outlined polygon.

**ALL**

displays an outlined, filled polygon.
(display-options)
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or both of the following options:

OUTLINE
displays an outlined polygon

FILL
displays a filled polygon

Default STANDARD

Tip Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the polygon.

DRAWSPACE=GRAPHPERCENT | GRAPHPXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPXEL |
| DATAVALUE
specifies a default drawing space and drawing units for this BEGINPOLYGON block.

Default LAYOUTPERCENT

Interactions This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this BEGINPOLYGON statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the XSPACE= and YSPACE= options.

See “About the Drawing Space and Drawing Units” on page 1322

FILLATTRS=style-element | style-element (fill-options) | (fill-options)
specifies the fill attributes for this BEGINPOLYGON block.

Default The GraphAnnoShape style element

Tip The TRANSPARENCY= option sets the transparency for the fill and the outline. You can combine this option with TRANSPARENCY= to set one transparency for the outlines but a different transparency for the fill. Example:

transparency=0.2 fillattrs=(transparency=0.6)

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Fill Color Options” on page 1520 for available fill-options.

LAYER=FRONT | BACK
specifies the layer on which this BEGINPOLYGON block’s output is drawn.

FRONT
draws the output on top of the graph.
BACK

draws the output behind the background areas, such as a layout or legend
background.

Default  FRONT

Tip  For elements that are obstructed because they are in the back layer,
you can suppress the display of filled areas in the graph. You can also
use transparency to manage the element visibility. For more information,
see “About Drawing Layers” on page 1325.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)

specifies the outline attributes for this BEGINPOLYGON block.

Default  The GraphAnnoShape style element.

Interaction  For this option to have any effect, the outline must be enabled by the
DISPLAY= option.

See  “General Syntax for Attribute Options” on page 1519 for the syntax
on using a style-element.

“Line Options” on page 1522 for available line-options.

TRANSPARENCY=number

specifies the degree of the transparency of this BEGINPOLYGON block’s output.

Default  0

Range  0 (opaque) to 1 (entirely transparent)

Tip  The FILLATTRS option can be used to set transparency for just the
polygon’s filled area. You can combine this option with FILLATTRS= to
set one transparency for the outlines but a different transparency for the
fill. Example:

transparency=0.2 fillattrs=(transparency=0.6)

URL=“string” | string-expression

specifies an HTML page that is displayed when the output of this draw statement
is selected.

Note: This option is valid starting in SAS 9.4M1.

“string”

string-expression

specifies a valid HTML page reference (HREF) for the graphical element that
is drawn by this draw statement.

Example  http://www.sas.com

Requirement  To generate selectable graphical elements, you must include an
ODS GRAPHICS ON statement that specifies the IMAGEMAP
option, and you must write the output to the ODS HTML
destination.

Tip  By default, drill-down links open in a new browser window (link
target _blank). To specify a different target for your drill-down links,
such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**

specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

**XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**

specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

Default: The setting that is in effect for the DRAWSPACE= option.

**YAXIS=Y | Y2**

specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

**YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**

specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default: The setting that is in effect for the DRAWSPACE= option.

**DRAW Statement Optional Arguments**

The following options can be used in the DRAW statement.

<table>
<thead>
<tr>
<th>Statement Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAWSPACE</td>
<td>Specifies a default drawing space and drawing units for the drawn lines.</td>
</tr>
<tr>
<td>XSPACE</td>
<td>Specifies the drawing space and drawing units for interpreting the X value.</td>
</tr>
<tr>
<td>YSPACE</td>
<td>Specifies the drawing space and drawing units for interpreting the Y value.</td>
</tr>
</tbody>
</table>
Details

Statement Description

A polygon is built by using the BEGINPOLYGON statement to specify the polygon’s starting X,Y coordinate, and then specifying the remaining points by nesting a series of DRAW statements (see "DRAW Statement" on page 1335) within the BEGINPOLYGON block. The block is closed with an ENDPOLYGON statement. To manage the location and drawing units for the polygon, you can use the XAXIS=, YAXIS=, XSPACE=, and YSPACE= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, "Key Concepts for Using Draw Statements," on page 1319. For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

DRAW Statement

The nested DRAW statements within a BEGINPOLYGON block specify a series of points for a polygon. Each DRAW statement draws a straight line from the previous point to the endpoint that is specified in the DRAW statement’s X and Y arguments. The first DRAW statement starts its line from the X,Y point that is specified in the BEGINPOLYGON statement.

You can specify as many DRAW statements as needed to complete the polygon. If the last DRAW statement does not end at the first point in the polygon (specified in BEGINPOLYGON), then a line is automatically generated to close the polygon shape. For a specific example, see the "Example Program" on page 1336.

Example: BEGINPOLYGON Statement

The following graph was generated by the "Example Program" on page 1336. The example uses a BEGINPOLYGON block to draw a polygon around the Setosa species of Iris in a plot that shows petal sizes for three Iris species. The DRAWSPACE= option in the BEGINGRAPH statements specifies that the polygon be drawn in the data space. The BEGINPOLYGON statement specifies the starting X,Y point. For the BEGINPOLYGON options, DISPLAY= displays only the fill for the polygon, TRANSPARENCY= adds a degree of transparency to the fill, and FILLATTRS= sets the fill color to yellow. The example also uses DRAWARROW and DRAWTEXT statements to draw an annotation for the polygon.
Example Program

proc template;
    define statgraph discretelegend;
        begingraph / drawspace=datavalue;
            entrytitle "Iris Petal Dimensions";
            layout overlayequated / equatetype=equate;
                scatterplot x=petallength y=petalwidth / group=species name="s";
                ellipse x=petallength y=petalwidth / type=predicted alpha=.2
                    name="p80" legendlabel="80%" outlineattrs=graphconfidence;
                ellipse x=petallength y=petalwidth / type=predicted alpha=.05
                    name="p95" legendlabel="95%" outlineattrs=graphconfidence2;
                beginpolygon x=9 y=2 /display=(fill)
                    fillAttrs=(color=yellow transparency=0.75);
                draw x=13 y=5;
                draw x=16 y=7;
                draw x=17 y=6;
                draw x=20 y=5;
                draw x=20 y=1;
                draw x=17 y=1;
                draw x=15 y=0;
                draw x=14 y=0;
                draw x=11 y=0;
                endpolygon;
            drawtext textattrs=(size=11pt) "Setosa" /
                x=33 y=0 width=15 anchor=top;
            drawarrow x1=28 y1=-1 x2=19 y2=2 /
                lineattrs=(thickness=1px)
                    arrowheadshape=barbed arrowheadscale=0.1;
            discretelegend "s" / title="Species:";
    endgraph;
enddefine;
endtemplate;
BEGINPOLYLINE Statement

Defines the starting point for drawing a polyline in the graph.

Syntax

BEGINPOLYLINE
X=constant | scalar-expression
Y=constant | scalar-expression </options>;

DRAW
X=constant | scalar-expression
Y=constant | scalar-expression </options>;

<... more DRAW statements ...>
ENDPOLYLINE;

Summary of Optional Arguments

Appearance options

LAYER=FRONT | BACK
specifies the layer on which this BEGINPOLYLINE block’s output is drawn.

LINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the line attributes for this BEGINPOLYLINE block.

TRANSPARENCY=number
specifies the degree of the transparency of the BEGINPOLYLINE block’s output.

Axes options

DISCRETEOFFSET=number
specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

XAXIS=X | X2
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

YAXIS=Y | Y2
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Drawing space options
specifies a default drawing space and drawing units for this BEGINPOLYLINE block.

specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

specifies an HTML page that is displayed when the output of this draw statement is selected.

Required Arguments

The following options are required in the BEGINPOLYLINE statement and the DRAW statement.

specifies the X value for the starting point in the polyline.

The value that is set for this argument is interpreted using the XSPACE= option. When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

When XSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

specifies the Y value for the starting point in the polyline.

The value that is set for this argument is interpreted using the YSPACE= option. When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

When YSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

Optional Arguments

The following options can be used in the BEGINPOLYLINE statement.

specifies an amount to offset from the discrete X values, or the discrete Y values, or both.
**DRAWSPACE=**

- **GRAPHPERCENT**
- **GRAPHPIXEL**
- **LAYOUTPERCENT**
- **LAYOUTPIXEL**
- **WALLPERCENT**
- **WALLPIXEL**
- **DATAPERCENT**
- **DATAPIXEL**
- **DATAVALUE**

Specifies a default drawing space and drawing units for this BEGINPOLYLINE block.

**Default**

- **LAYOUTPERCENT**

**Interactions**

This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this BEGINPOLYLINE statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the XSPACE= and YSPACE= options.

**See**

- "About the Drawing Space and Drawing Units" on page 1322

**LAYER=**

- **FRONT**
- **BACK**

Specifies the layer on which this BEGINPOLYLINE block's output is drawn.

**FRONT**

Draws the output on top of the graph.

**BACK**

Draws the output behind the background areas, such as a layout or legend background.

**Default**

- **FRONT**

**Tip**

For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see "About Drawing Layers" on page 1325.
The GraphAnnoLine style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**TRANSPARENCY=number**
specifies the degree of the transparency of the BEGINPOLYLINE block’s output.

**Default** 0

**Range** 0 (opaque) to 1 (entirely transparent)

**URL=“string” | string-expression**
specifies an HTML page that is displayed when the output of this draw statement is selected.

---

**Note:** This option is valid starting in SAS 9.4M1.

---

“string” string-expression
specifies a valid HTML page reference (HREF) for the graphical element that is drawn by this draw statement.

**Example** http://www.sas.com

**Requirement** To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Tip** By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET=“target” option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

**Default** X

**Interaction** This option has effect only if XSPACE=DATAVALUE.

**XSPACE=GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

**Default** The setting that is in effect for the DRAWSPACE= option.

**YAXIS=Y | Y2**
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.
YSPACE=

This option has effect only if YSPACE=DATAVALUE.

specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default 

The setting that is in effect for the DRAWSPACE= option.

DRAW Statement Optional Arguments

The following options can be used in the DRAW statement.

<table>
<thead>
<tr>
<th>Statement Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAWSPACE</td>
<td>Specifies a default drawing space and drawing units for the drawn lines.</td>
</tr>
<tr>
<td>XSPACE</td>
<td>Specifies the drawing space and drawing units for interpreting the X value.</td>
</tr>
<tr>
<td>YSPACE</td>
<td>Specifies the drawing space and drawing units for interpreting the Y value.</td>
</tr>
</tbody>
</table>

Details

Statement Description

A polyline is built by using the BEGINPOLYLINE statement to specify the polyline’s starting X,Y coordinate, and then specifying the remaining points by nesting a series of DRAW statements (see “DRAW Statement” on page 1341 ) within the BEGINPOLYLINE block. The block is closed with an ENDPOLYLINE statement. To manage the location and drawing units for the polyline, you can use the XAXIS=, YAXIS=, XSPACE=, and YSPACE= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, "Key Concepts for Using Draw Statements," on page 1319 . For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

DRAW Statement

The nested DRAW statements within a BEGINPOLYLINE block specify a series of points for a polyline. Each DRAW statement draws a straight line from the previous point to the endpoint that is specified in the DRAW statement’s X and Y arguments. The first DRAW statement starts its line from the X,Y point that is specified in the BEGINPOLYLINE statement. You can specify as many DRAW statements as needed to complete the polyline.

For a specific example, see the “Example Program” on page 1342.
Example: BEGINPOLYLINE Statement

The following graph was generated by the "Example Program" on page 1342. The example uses two BEGINPOLYLINE blocks to highlight two student data points. The DRAWSPACE= option in the BEGINGRAPH statements specifies that the polyline be drawn in the data space. The BEGINPOLYLINE statements specify the starting X,Y points for two polyline, and the DRAW statements complete the lines. The example also uses two DRAWTEXT statements to label the data points of interest.

Example Program

```plaintext
proc template;
    define statgraph drawoval;
        begingraph / drawspace=datavalue;
            entrytitle "Regression Fit Plot";
            layout overlay;
                modelband "myclm";
                scatterplot x=height y=weight;
                regressionplot x=height y=weight / alpha=0.01 clm="myclm";
                drawtext "Alfred" / x=69 y=112 anchor=top;
                drawtext "Barbara" / x=65.4 y=97 anchor=top width=15;
                beginpolyline x=69 y=105;
                    draw x=69 y=85;
                    draw x=65.3 y=85;
                    draw x=65.3 y=90;
                endpolyline;
                beginpolyline x=67 y=85;
                    draw x=67 y=78;
                endpolyline;
                drawtext "New in Class" / x=67 y=77 anchor=top width=15;
        endlayout;
    endlayout;
```
DRAWORD Statement

Draws an arrow (a directed line segment) from one point to another point.

Syntax

**DRAWORD**  
\[ X1=constant \mid scalar-expression \]  
\[ Y1=constant \mid scalar-expression \]  
\[ X2=constant \mid scalar-expression \]  
\[ Y2=constant \mid scalar-expression \]  
\[ \langle \text{options} \rangle; \]

Summary of Optional Arguments

Appearance options

- **LAYER=**FRONT | BACK  
  specifies the layer on which this draw statement's output is drawn.

- **LINEATTRS=**style-element | style-element (line-options) | (line-options)  
  specifies the line attributes for this draw statement.

- **TRANSPARENCY=**number  
  specifies the degree of the transparency of this draw statement's output.

Arrow options

- **ARROWHEADDIRECTION=**OUT | IN | BOTH  
  specifies the direction of the arrowhead(s) at the end(s) of the arrow shaft.

- **ARROWHEADSCALE=**positive-number  
  specifies an arrowhead scale factor based on the thickness of the arrow shaft.

- **ARROWHEADSHAPE=**OPEN | CLOSED | FILLED | BARBED  
  specifies the shape of the arrowhead(s).

Axes options

- **DISCRETEOFFSET=**number  
  specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

- **XAXIS=**X | X2  
  specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

- **YAXIS=**Y | Y2  
  specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.
Drawing space options

**DRAWSPACE=**GRAPHPERCENT | GRAPHPXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPXEL | DATAVALUE

specifies a default drawing space and drawing units for this draw statement.

**X1SPACE=**GRAPHPERCENT | GRAPHPXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPXEL | DATAVALUE

specifies the drawing space and drawing units for interpreting the X1 value.

**X2SPACE=**GRAPHPERCENT | GRAPHPXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPXEL | DATAVALUE

specifies the drawing space and drawing units for interpreting the X2 value.

**Y1SPACE=**GRAPHPERCENT | GRAPHPXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPXEL | DATAVALUE

specifies the drawing space and drawing units for interpreting the Y1 value.

**Y2SPACE=**GRAPHPERCENT | GRAPHPXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPXEL | DATAVALUE

specifies the drawing space and drawing units for interpreting the Y2 value.

**ODS options**

**URL="string" | string-expression**

specifies an HTML page that is displayed when the output of this draw statement is selected.

**Required Arguments**

**X1=**constant | scalar-expression

specifies the X value of one arrow-shaft endpoint.

**Y1=**constant | scalar-expression

specifies the Y value of one arrow-shaft endpoint.
Note: When $Y1SPACE=\text{DATAVALUE}$ and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

**X2=constant | scalar-expression**
specifies the X value of one arrow-shaft endpoint.

**Interactions**
This value that is set for this option is interpreted using the $X2SPACE=$ option.

When $X2SPACE=\text{DATAVALUE}$, the value is interpreted using the $XAXIS=$ option.

Note: When $X2SPACE=\text{DATAVALUE}$ and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

**Y2=constant | scalar-expression**
specifies the Y value of one arrow-shaft endpoint.

**Interactions**
This value that is set for this option is interpreted using the $Y2SPACE=$ option.

When $Y2SPACE=\text{DATAVALUE}$, the value is interpreted using the $YAXIS=$ option.

Note: When $Y2SPACE=\text{DATAVALUE}$ and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

Optional Arguments

**ARROWHEADDIRECTION=OUT | IN | BOTH**
specifies the direction of the arrowhead(s) at the end(s) of the arrow shaft.

OUT
specifies a single arrowhead drawn at $(X2,Y2)$ and pointing away from $(X1,Y1)$

IN
specifies a single arrowhead drawn at $(X1,Y1)$ and pointing away from $(X2,Y2)$

BOTH
specifies two arrowheads, one at the IN position and one at the OUT position

Default: OUT

Tip: Use the ARROWHEADSHAPE= option to control the arrowhead appearance.

**ARROWHEADSHAPE=OPEN | CLOSED | FILLED | BARBED**
specifies the shape of the arrowhead(s). The following figure shows the arrowhead shapes.
Tip Use the ARROWHEADDIRECTION= option to control the arrow direction.

**ARROWHEADSCALE=positive-number**
specifies an arrowhead scale factor based on the thickness of the arrow shaft.

Default 1.0

Restriction The minimum size for arrowheads is 8 pixels. If you specify a value for ARROWHEADSCALE= that scales the arrowhead below 8 pixels, an 8-pixel arrowhead is used instead. There is no restriction on the maximum size.

Tip Use a factor greater than 1.0 to make a larger arrowhead.

**DISCRETEOFFSET=number**
specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

Default 0 (no offset, output is centered on the discrete X values, or the discrete Y values, or both)

Range −0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the right. If the Y axis is discrete, then a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.

Restriction This option applies only when the options X1SPACE=, X2SPACE=, Y1SPACE=, or Y2SPACE= use DATAVALE, and when X1, X2, Y1, or Y2 are values on a discrete axis. For nondiscrete axes, this option is ignored.

Tip Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**DRAWSPACE=GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE
**specifies a default drawing space and drawing units for this draw statement.

Default LAYOUTPERCENT

Interactions This statement and all of the draw statements inherit the global DRAWSSPACE= setting from the DRAWSPACE= option in the
BEGINGRAPH statement. Setting this option changes the setting for only this draw statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options X1SPACE=, Y1SPACE=, X2SPACE=, or Y2SPACE=.

See “About the Drawing Space and Drawing Units” on page 1322

**LAYER=** FRONT | BACK

specifies the layer on which this draw statement’s output is drawn.

- **FRONT**
  draws the output on top of the graph.

- **BACK**
  draws the output behind the background areas, such as a layout or legend background.

Default FRONT

**Tip**
For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 1325.

**LINEATTRS=** style-element | style-element (line-options) | (line-options)

specifies the line attributes for this draw statement.

Default The GraphAnnoLine style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**TRANSPARENCY=** number

specifies the degree of the transparency of this draw statement’s output.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

**URL=** "string" | string-expression

specifies an HTML page that is displayed when the output of this draw statement is selected.

Note: This option is valid starting in SAS 9.4M1.

"string"

string-expression

specifies a valid HTML page reference (HREF) for the graphical element that is drawn by this draw statement.

Example http://www.sas.com

Requirement To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP
**Tip**

By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

XAXIS=X | X2

specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

Default X

Interaction This option has effect only if X1SPACE=DATAVALUE.

X1SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE

specifies the drawing space and drawing units for interpreting the X1 value.

Default The setting that is in effect for the DRAWSPACE= option.

Interaction This option overrides the DRAWSPACE= setting only for the X1 value.

X2SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE

specifies the drawing space and drawing units for interpreting the X2 value.

Default The setting that is in effect for the DRAWSPACE= option.

Interaction This option overrides the DRAWSPACE= setting only for the X2 value.

YAXIS=Y | Y2

specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default Y

Interaction This option has effect only if Y1SPACE=DATAVALUE.

Y1SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE

specifies the drawing space and drawing units for interpreting the Y1 value.

Default The setting that is in effect for the DRAWSPACE= option.

Interaction This option overrides the DRAWSPACE= setting only for the Y1 value.
Y2SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the Y2 value

Default          The setting that is in effect for the DRAWSPACE= option.
Interaction      This option overrides the DRAWSPACE= setting only for the Y2 value.

Details
A DRAWARROW statement draws a line (arrow shaft) from a specified starting point (X1,Y1) to a specified ending point (X2,Y2). It also displays an arrowhead at either or both ends of the line. DRAWARROW is similar to a DRAWLINE statement, using many of the same options, but it has additional options for controlling the arrowhead(s).

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, “Key Concepts for Using Draw Statements,” on page 1319. For detailed usage information, consult the SAS Graph Template Language: User's Guide.

Example: DRAWARROW Statement
The following graph was generated by the “Example Program” on page 1350. The example shows a common application of a DRAWARROW and DRAWTEXT statements to identify a specific part of the graph and add explanatory text.
Example Program

```sas
proc template;
   define statgraph arrow;
      begingraph;
         entrytitle "Microsoft Stock Prices between 2000 and 2002";
         layout overlay;
            seriesplot x=date y=close;
            drawarrow x1="01NOV2001"d y1=75 x2="01NOV2001"d y2=64.21 /
               x1space=datavalue y1space=wallpercent
               x2space=datavalue y2space=datavalue
               arrowheadshape=filled lineattrs=(color=red) ;
            drawtext "Introduction of Windows XP" / width=25 anchor=bottom
               border=true borderattrs=(color=red)
               x="01NOV2001"d y=75 xspace=datavalue yspace=wallpercent;
         endlayout;
      endgraph;
   end;

proc sgrender data=sashelp.stocks template=arrow;
   where stock="Microsoft" and Year(date) between 2000 and 2002;
run;
```

**DRAWIMAGE Statement**

Draws an image in the graph.

**Syntax**

```
DRAWIMAGE "image-file-spec" </options>;
```

**Summary of Optional Arguments**

**Appearance options**

- **BORDER=TRUE | FALSE**
  specifies whether a border is drawn around the image.

- **BORDERATTRS=style-element | style-element (line-options) | (line-options)**
  specifies the border line attributes for this draw statement.

- **LAYER=FRONT | BACK**
  specifies the layer on which this draw statement's output is drawn.

- **ROTATE=number**
  specifies the angle of rotation for the image, measured in degrees.

- **TRANSPARENCY=number**
  specifies the degree of the transparency of this draw statement's output.

**Axes options**

- **DISCRETEOFFSET=number**
  specifies an amount to offset from the discrete X values, or the discrete Y values, or both.
**XAXIS=X | X2**
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

**YAXIS=Y | Y2**
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

**Drawing space options**

**DRAWSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies a default drawing space and drawing units for this draw statement.

**XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

**YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

**Location options**

**ANCHOR=CENTERTOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT**
specifies an anchor point for the image.

**X=constant | scalar-expression**
specifies the anchor point’s X coordinate.

**Y=constant | scalar-expression**
specifies the anchor point’s Y coordinate.

**ODS options**

**URL="string" | string-expression**
specifies an HTML page that is displayed when the output of this draw statement is selected.

**Size options**

**HEIGHT=positive-number**
specifies the height of the image’s bounding box.

**HEIGHTUNIT=PERCENT | PIXEL | DATA**
specifies whether the positive-number that is specified in the HEIGHT= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

**SCALE=FIT | FITHEIGHT | FITWIDTH | TILE**
specifies how the image is scaled within the bounding box.

**SIZEUNIT=PERCENT | PIXEL | DATA**
specifies whether the default units for the size of the image’s bounding box are percentage values, or pixel values, or values that are in the unit of the data.
WIDTH=\texttt{positive-number}

specifies the width of the image’s bounding box.

\textbf{WIDTHUNIT=\texttt{PERCENT | PIXEL | DATA}}

specifies whether the \texttt{positive-number} that is specified in the \texttt{WIDTH=} option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

\textbf{Required Argument}

\textit{image-file-spec}

specifies the name, image type, and location of the image. The \textit{image-file-spec} value must be enclosed in double quotation marks and must be specified as a local, physical file path (for example, "c:\temp\saslogo.gif"). The supported image types are GIF, JPEG, and PNG raster or bitmap format.

\textbf{Restrictions}

URL access to image files is not supported. The image file must exist on the file system.

In SAS 9.4 and earlier releases, you cannot use a dynamic variable for the \textit{image-file-spec} value. This restriction is removed starting with \textbf{SAS 9.4M1}.

\textbf{Optional Arguments}

\textbf{ANCHOR=}\texttt{CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT}

specifies an anchor point for the image. The anchor point is relative to the unrotated image. The anchor point can be at the center of the image or at eight points on the border of the image bounding box. The following figure shows the anchor points for TOPLEFT and LEFT.

The coordinates of the anchor point are set by the \texttt{X=} and \texttt{Y=} options, and by the \texttt{XSPACE=} and \texttt{YPSPACE=} options. The \texttt{XAXIS=} and \texttt{YAXIS=} option might affect positioning when the \texttt{XSPACE=} or \texttt{YPSPACE=} options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

The image has a fixed height and a fixed width, determined by the \texttt{HEIGHT=}, \texttt{HEIGHTUNIT=}, \texttt{WIDTH=} and \texttt{WIDTHUNIT=} options. The height of the text grows in a direction that is related to the anchor point. For example, if \texttt{ANCHOR=TOPLEFT}, then the image height extends downward from the anchor point and its width extends to the right. If \texttt{ANCHOR=CENTER}, then half the image width and half the image height extend equally left and right, as well as top to bottom from the anchor point. If \texttt{ANCHOR=BOTTOM}, then the image height extends upward from the anchor point and the image width is centered at the anchor point.

When the image is rotated, the anchor point remains relative to the unrotated image while the image is rotated on its anchor point. See \texttt{ROTATE=} on page 1354.
**BORDER=TRUE | FALSE**
specifies whether a border is drawn around the image.

Default: FALSE

**Tip**
Use the BORDERATTRS= option to control the appearance of the border.

**See**
“boolean” on page 1479 for other Boolean values that you can use.

**BORDERATTRS=style-element | style-element (line-options) | (line-options)**
specifies the border line attributes for this draw statement.

Default: The GraphBorderLines style element.

**Interaction**
BORDER=TRUE must be set for this option to have any effect.

**See**
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

**DISCRETEOFFSET=number**
specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

Default: 0 (no offset, output is centered on the discrete X values, or the discrete Y values, or both)

**Range**
−0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the right. If the Y axis is discrete, then a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.

**Restriction**
This option applies only when the options XSPACE= or YSPACE= use DATAVALUE, and when X or Y are values on a discrete axis. For nondiscrete axes, this option is ignored.

**Tip**
Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**DRAWSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies a default drawing space and drawing units for this draw statement.

Default: LAYOUTPERCENT

**Interactions**
This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the
BEGINGRAPH statement. Setting this option changes the setting for only this draw statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT=, or WIDTHUNIT=.

See “About the Drawing Space and Drawing Units” on page 1322

**HEIGHT=** *positive-number*

specifies the height of the image's bounding box.

Default The height of the image.

Interaction The interpretation of this height setting is determined by the combined settings of the HEIGHTUNIT= and YSPACE= options.

**HEIGHTUNIT=** PERCENT | PIXEL | DATA

specifies whether the *positive-number* that is specified in the HEIGHT= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default PERCENT

Interaction This setting combines with the YSPACE= setting to interpret the height that is set in the HEIGHT= option.

**LAYER=** FRONT | BACK

specifies the layer on which this draw statement's output is drawn.

**FRONT**

draws the output on top of the graph.

**BACK**

draws the output behind the background areas, such as a layout or legend background.

Default FRONT

Tip For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 1325.

**ROTATE=** *number*

specifies the angle of rotation for the image, measured in degrees. The image is rotated around its anchor point. The angle is measured from a horizontal line passing through the anchor point of the image to the right. The following figure shows the rotation of an image around a top anchor point.
Positive angles rotate the image counter clockwise, and negative angles rotate the image clockwise. The angle specification can exceed 360 degrees in absolute value.

**Default**

0. No rotation is performed.

**Note**

If you specify the SVG format for your graph and your graph includes a rotated image, then the graph is rendered as a bitmapped image rather than an SVG image. In that case, a note is written to the SAS log indicating the change in format.

**See**

ANCHOR= on page 1352

**SCALE**=FIT | FITHEIGHT | FITWIDTH | TILE

specifies how the image is scaled within the bounding box.

**FIT**

scales the image to fit the bounding box.

Aspect ratio is not maintained.

**FITHEIGHT**

scales the image to fit the height of the bounding box.

The width is computed from the height and the image's aspect ratio.

**FITWIDTH**

scales the image to fit the width of the bounding box.

The height is computed from the width and the image’s aspect ratio.

**TILE**

tiles the image as needed to fit the bounding image.

The last tile in a row or column might be clipped by the bounding box.

**Default**

FIT

**SIZEUNIT**=PERCENT | PIXEL | DATA

specifies whether the default units for the size of the image’s bounding box are percentage values, or pixel values, or values that are in the unit of the data.

**Default**

PERCENT

**Interaction**

If the HEIGHTUNIT= or WIDTHUNIT= option is also used, then it overrides this option for that dimension.
**TRANSPARENCY=number**
specifies the degree of the transparency of this draw statement's output.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

Note This option also affects the image border.

**URL="string" | string-expression**
specifies an HTML page that is displayed when the output of this draw statement is selected.

Note: This option is valid starting in **SAS 9.4M1**.

"string"
string-expression
specifies a valid HTML page reference (HREF) for the graphical element that is drawn by this draw statement.

Example http://www.sas.com

Requirement To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Tip By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See **DRILLTARGET=** on page 1489.

**WIDTH=positive-number**
specifies the width of the image’s bounding box.

Default The width of the image.

**WIDTHUNIT=PERCENT | PIXEL | DATA**
specifies whether the *positive-number* that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default PERCENT

Interaction This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= option.

**X=constant | scalar-expression**
specifies the anchor point’s X coordinate.

Default 50

Interactions The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the XSPACE= option.
If XSPACE=DATAVALUE, then this option's value is interpreted using the XAXIS= option.

**Note**
When XSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

**XAXIS=X | X2**
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

Default **X**

Interaction This option has effect only if XSPACE=DATAVALUE.

**XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

Default The setting that is in effect for the DRAWSPACE= option.

**Y=constant | scalar-expression**
specifies the anchor point’s Y coordinate.

Default **50**

Interactions The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the YSPACE= option.

If YSPACE=DATAVALUE, then this option's value is interpreted using the YAXIS= option.

**Note**
When YSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

**YAXIS=Y | Y2**
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default **Y**

Interaction This option has effect only if YSPACE=DATAVALUE.

**YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default The setting that is in effect for the DRAWSPACE= option.
Details

A DRAWIMAGE statement draws an image in a graph. By default, the image is drawn in the center of the graph. You can change the default position with the options ANCHOR=, X=, Y=, XSPACE=, and YSPACE=. By default, the image is drawn in the actual image size. You can change the default size with the WIDTH= and HEIGHT= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, “Key Concepts for Using Draw Statements,” on page 1319. For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

Example: DRA WIMAGE Statement

Example Graph

The following graph was generated by the “Example Program” on page 1358. The example shows how to display an image in the bottom right corner of the graph wall.

Example Program

```
proc template;
  define statgraph image;
  begingraph;
    entrytitle "Regression Fit Plot";
    layout overlay;
      modelband "myclm";
      scatterplot x=height y=weight / primary=true;
  endgraph;
end;
```
regressionplot x=height y=weight / alpha=0.01 clm="myclm";
drawimage "c:\temp\saslogo.gif" /
   anchor=bottomright x=98 y=2
drawspace=wallpercent ;
endlayout;
endgraph;
end;

proc sgrender data=sashelp.class template=image;
run;

**DRAWLINE Statement**

Draws a line from one point to another point.

**Syntax**

```plaintext
DRAWLINE X1=constant | scalar-expression
   Y1=constant | scalar-expression
   X2=constant | scalar-expression
   Y2=constant | scalar-expression <options>;
```

**Summary of Optional Arguments**

**Appearance options**

- **LAYER=** FRONT | BACK
  - specifies the layer on which this draw statement's output is drawn.
- **LINEATTRS=** style-element | style-element (line-options) | (line-options)
  - specifies the line attributes for this draw statement.
- **TRANSPARENCY=** number
  - specifies the degree of the transparency of this draw statement's output.

**Axes options**

- **DISCRETEOFFSET=** number
  - specifies an amount to offset from the discrete X values, or the discrete Y values, or both.
- **XAXIS=** X | X2
  - specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.
- **YAXIS=** Y | Y2
  - specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

**Drawing space options**

```plaintext
DRAWSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
```
specifies a default drawing space and drawing units for this draw statement.

\[X1\text{SPACE}=\text{GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE}\]

specifies the drawing space and drawing units for interpreting the X1 value.

\[X2\text{SPACE}=\text{GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE}\]

specifies the drawing space and drawing units for interpreting the X2 value.

\[Y1\text{SPACE}=\text{GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE}\]

specifies the drawing space and drawing units for interpreting the Y1 value.

\[Y2\text{SPACE}=\text{GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE}\]

specifies the drawing space and drawing units for interpreting the Y2 value.

**ODS options**

\[\text{URL}="\text{string} | \text{string-expression}\]

specifies an HTML page that is displayed when the output of this draw statement is selected.

**Required Arguments**

\[\text{X1=}\text{constant | scalar-expression}\]

specifies the X value of the starting point for the line.

**Interactions**

This value that is set for this option is interpreted using the X1SPACE= option.

When X1SPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

**Note**

When X1SPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

\[\text{Y1=}\text{constant | scalar-expression}\]

specifies the Y value of the starting point for the line.

**Interactions**

This value that is set for this option is interpreted using the Y1SPACE= option.

When Y1SPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

**Note**

When Y1SPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.
**X2=constant | scalar-expression**
specifies the X value of the endpoint for the line.

---

**Interactions**
This value that is set for this option is interpreted using the X2SPACE= option.

When X2SPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

---

**Note**
When X2SPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

---

**Y2=constant | scalar-expression**
specifies the Y value of the endpoint for the line.

---

**Interactions**
This value that is set for this option is interpreted using the Y2SPACE= option.

When Y2SPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

---

**Note**
When Y2SPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

---

**Optional Arguments**

**DISCRETEOFFSET=number**
specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

---

**Default**
0 (no offset, output is centered on the discrete X values, or the discrete Y values, or both)

---

**Range**
–0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the right. If the Y axis is discrete, then a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.

---

**Restriction**
This option applies only when the options X1SPACE=, X2SPACE=, Y1SPACE=, or Y2SPACE= use DATAVALUE, and when X1, X2, Y1, or Y2 are values on a discrete axis. For nondiscrete axes, this option is ignored.

---

**Tip**
Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

---

**DRAWSPACE=GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies a default drawing space and drawing units for this draw statement.
This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this draw statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options X1SPACE=, Y1SPACE=, X2SPACE=, or Y2SPACE=.

See “About the Drawing Space and Drawing Units” on page 1322

**LAYER=FRONT | BACK**

specifies the layer on which this draw statement’s output is drawn.

- **FRONT**
  - draws the output on top of the graph.

- **BACK**
  - draws the output behind the background areas, such as a layout or legend background.

Default **FRONT**

**Tip**

For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 1325.

**LINEATTRS=**

specifies the line attributes for this draw statement.

- **style-element**
- **style-element (line-options)**
- **(line-options)**

Default The GraphAnnoLine style element.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a **style-element**.

“Line Options” on page 1522 for available **line-options**.

**TRANSPARENCY=**

specifies the degree of the transparency of this draw statement’s output.

Default **0**

Range **0–1**, where 0 is opaque and 1 is entirely transparent

**URL=**

specifies an HTML page that is displayed when the output of this draw statement is selected.

**Tip**

This option is valid starting in **SAS 9.4M1**.

**string**

**string-expression**

specifies a valid HTML page reference (HREF) for the graphical element that is drawn by this draw statement.
Example  http://www.sas.com

Requirement  To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Tip  By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**XAXIS=X | X2**
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

Default  X

Interaction  This option has effect only if X1SPACE=DATAVALUE.

**X1SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the X1 value.

Default  The setting that is in effect for the DRAWSPACE= option.

Interaction  This option overrides the DRAWSPACE= setting only for the X1 value.

**X2SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the X2 value.

Default  The setting that is in effect for the DRAWSPACE= option.

Interaction  This option overrides the DRAWSPACE= setting only for the X2 value.

**YAXIS=Y | Y2**
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default  Y

Interaction  This option has effect only if Y1SPACE=DATAVALUE.

**Y1SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the Y1 value.

Default  The setting that is in effect for the DRAWSPACE= option.
Interaction

This option overrides the DRAWSPACE= setting only for the Y1 value.

**Y2SPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**

specifies the drawing space and drawing units for interpreting the Y2 value

**Default**

The setting that is in effect for the DRAWSPACE= option.

Interaction

This option overrides the DRAWSPACE= setting only for the Y2 value.

**Details**

A DRAWLINE statement draws a line from a starting point that is specified with the X1 and Y1 arguments, to an ending point that is specified with the X2 and Y2 arguments. DRAWLINE is similar to a DRAWARROW statement, using many of the same options, but without the options for controlling the arrowhead(s).

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, "Key Concepts for Using Draw Statements," on page 1319. For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

**Example: DRAWLINE Statement**

The following graph was generated by the “Example Program” on page 1365. The example shows how to draw a diagonal reference line. One endpoint is point 0,0 and the other is point 100,100. Both points are specified in the WALL area with PERCENT units, making it easy to position the line without regard to the axis data ranges or the axis offsets. To draw the line behind the series line and grid lines, you can set LAYER= BACK and use the parent layout statement to turn off the display of the wall.
Example Program

```plaintext
proc template;
  define statgraph diagonal;
  begingraph;
    entrytitle "Open vs. Close Price for Intel Stock 2003";
    layout overlay / walldisplay=(outline)
      xaxisopts=(griddisplay=on)
      yaxisopts=(griddisplay=on);
    seriesplot x=open y=close / datalabel=date;
    drawline x1=0 y1=0 x2=100 y2=100 /
      x1space=wallpercent y1space=wallpercent
      x2space=wallpercent y2space=wallpercent
      lineattrs=GraphReference layer=back;
    endlayout;
  endgraph;
end;

proc sgrender data=sashelp.stocks template=diagonal;
  where stock="Intel" and year(date)=2003;
  format date monname3.;
run;
```

DRAWOVAL Statement

Draws an oval in the graph.
Syntax

**DRAWOVAL**  
X=constant | scalar-expression  
Y=constant | scalar-expression  
WIDTH=constant | scalar-expression  
HEIGHT=constant | scalar-expression </options>;

Summary of Optional Arguments

**Appearance options**

DISPLAY=STANDARD | ALL | (display-options)  
Specifies the features to display for the oval.

FILLATTRS=style-element | style-element (fill-options) | (fill-options)  
Specifies the fill attributes for this draw statement.

LAYER=FRONT | BACK  
Specifies the layer on which this draw statement's output is drawn.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)  
Specifies the outline attributes for this draw statement.

ROTATE=number  
Specifies the angle of rotation for the oval, measured in degrees.

TRANSPARENCY=number  
Specifies the degree of the transparency of this draw statement's output.

**Axes options**

DISCRETEOFFSET=number  
Specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

XAXIS=X | X2  
Specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

YAXIS=Y | Y2  
Specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

**Drawing space options**

DRAWSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE  
Specifies a default drawing space and drawing units for this draw statement.

XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE  
Specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE  
Specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.
Location options
ANCHOR= CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT
specifies an anchor point for the oval.

ODS options
URL="string" | string-expression
specifies an HTML page that is displayed when the output of this draw statement is selected.

Size options
HEIGHTUNIT= PERCENT | PIXEL | DATA
specifies whether the positive-number that is specified in the HEIGHT= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
WIDTHUNIT= PERCENT | PIXEL | DATA
specifies whether the positive-number that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Required Arguments
X=constant | scalar-expression
specifies the X value of the anchor point.

Interactions The value that is set for this argument is interpreted using the XSPACE= option.
When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Note When XSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

Y=constant | scalar-expression
specifies the Y value of the anchor point.

Interactions The value that is set for this argument is interpreted using the YSPACE= option.
When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

Note When YSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

WIDTH=constant | scalar-expression
specifies the width of the oval.

Interactions The value that is set for this argument is interpreted using the WIDTHUNIT= and XSPACE= options.
When WIDTHUNIT=DATA, the value is interpreted using the XAXIS= option.
**HEIGHT=constant | scalar-expression**

specifies the height of the oval.

**Interactions**

The value that is set for this argument is interpreted using the
HEIGHTUNIT= and YSPACE= options.

When HEIGHTUNIT=DATA, the value is interpreted using the
YAXIS= option.

---

**Optional Arguments**

**ANCHOR=**

**CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT**

specifies an anchor point for the oval. The anchor point is relative to the
unrotated oval. The anchor point can be at the center of the oval or at eight
points on the bounding box of the rectangle. The following figure shows the
anchor points for TOPLEFT and CENTER.

The coordinates of the anchor point are set by the statement’s X and Y values,
and by the settings for the XSPACE= and YSPACE= options. The XAXIS= and
YAXIS= option might affect positioning when the XSPACE= or YSPACE= options
are set to DATAPIXEL, DATAPERCENT, or DATAVALE.

When the oval is rotated, the anchor point remains relative to the unrotated oval
while the oval is rotated on its anchor point. See **ROTATE=** on page 1370.

**Default**

CENTER

**DISCRETEOFFSET=**

**number**

specifies an amount to offset from the discrete X values, or the discrete Y
values, or both.

**Default**

0 (no offset, output is centered on the discrete X values, or the
discrete Y values, or both)

**Range**

−0.5 to +0.5, where 0.5 represents half the distance between
discrete ticks. If the X axis is discrete, then a positive offset is to the
right. If the Y axis is discrete, then a positive offset is up. If
REVERSE=TRUE on the X or Y axis, then the offset direction is also
reversed.

**Restriction**

This option applies only when the options XSPACE= or YSPACE=
use DATAVALE, and when X or Y are values on a discrete axis. For
nondiscrete axes, this option is ignored.

**Tip**

Setting the discrete offset for the plots does not affect the axis
minimum and maximum offsets. In some cases, setting a discrete
offset can cause clipping at each end of the axis. In those cases, use
the OFFSETMIN= and OFFSETMAX= axis options to increase the
axis minimum and maximum offsets to accommodate the discrete offset.

**DISPLAY=STANDARD | ALL | (display-options)**
specifies the features to display for the oval.

**STANDARD**
displays an outlined oval.

**ALL**
displays an outlined, filled oval.

**(display-options)**
specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or both of the following options:

**OUTLINE** displays an outlined oval

**FILL** displays a filled oval

**Default**: STANDARD

**Tip** Use the **OUTLINEATTRS= and FILLATTRS=** options to control the appearance of the oval.

**DRAWSPACE=GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies a default drawing space and drawing units for this draw statement.

**Default**: LAYOUTPERCENT

**Interactions**
This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this draw statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT=, or WIDTHUNIT=.

**See** "About the Drawing Space and Drawing Units" on page 1322

**FILLATTRS=** **style-element | style-element (fill-options) | (fill-options)**
specifies the fill attributes for this draw statement.

**Default**: The GraphAnnoShape style element Color attribute

**Tip**
The **TRANSPARENCY=** option sets the transparency for the fill and the outline. You can combine this option with TRANSPARENCY= to set one transparency for the outline but a different transparency for the fill.

Example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

**See** "General Syntax for Attribute Options" on page 1519 for the syntax on using a **style-element**.

"Fill Color Options" on page 1520 for available **fill-options**.
HEIGHTUNIT=PERCENT | PIXEL | DATA
specifies whether the positive-number that is specified in the HEIGHT= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default PERCENT

Interaction This setting combines with the YSPACE= setting to interpret the height that is set in the HEIGHT= option.

LAYER=FRONT | BACK
specifies the layer on which this draw statement’s output is drawn.

FRONT
draws the output on top of the graph.

BACK
draws the output behind the background areas, such as a layout or legend background.

Default FRONT

Tip For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 1325.

OUTLINEATTRS=style-element | style-element (line-options) | (line-options)
specifies the outline attributes for this draw statement.

Default The GraphAnnoLine style element.

Interaction For this option to have any effect, the outline must be enabled by the DISPLAY= option or by the ODS style that is in effect.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Line Options” on page 1522 for available line-options.

ROTATE=number
specifies the angle of rotation for the oval, measured in degrees. The angle is measured as if a horizontal line extended to the right through the oval anchor point as shown in the following figure.

<table>
<thead>
<tr>
<th>Rotation With ANCHOR=TOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Rotation</td>
</tr>
<tr>
<td>R0TATE=90</td>
</tr>
<tr>
<td>R0TATE=180</td>
</tr>
<tr>
<td>R0TATE=270</td>
</tr>
</tbody>
</table>
Positive angles rotate the oval counter clockwise, and negative angles rotate the oval clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0. No rotation is performed

See ANCHOR= on page 1368

**TRANSPARENCY=number**

specifies the degree of the transparency of this draw statement's output.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

Tip The FILLATTRS= option can be used to set transparency for just the filled area. You can combine this option with FILLATTRS= to set one transparency for the outline but a different transparency for the fill. Here is an example:

```plaintext
transparency=0.2 fillattrs=(transparency=0.6)
```

**URL= "string" | string-expression**

specifies an HTML page that is displayed when the output of this draw statement is selected.

---

**Example** http://www.sas.com

**Requirement** To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

**Tip** By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**WIDTHUNIT=PERCENT | PIXEL | DATA**

specifies whether the positive-number that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default PERCENT

Interaction This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= option.
**XAXIS=X | X2**
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

Default: X

Interaction: This option has effect only if XSPACE=DATAVALUE.

**XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

Default: The setting that is in effect for the DRAWSPACE= option.

**YAXIS=Y | Y2**
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default: Y

Interaction: This option has effect only if YSPACE=DATAVALUE.

**YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default: The setting that is in effect for the DRAWSPACE= option.

### Details

A DRAWOVAL statement draws an oval in a graph. The oval position is determined by the X and Y anchor points, and the size is determined by the HEIGHT and WIDTH settings. You can manage the oval position with the options ANCHOR=, XSPACE=, and YSPACE=. You can manage the oval size with the HEIGHTUNIT= and WIDTHUNIT= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, "Key Concepts for Using Draw Statements," on page 1319. For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

### Example: DRAWOVAL Statement

The following graph was generated by the “Example Program” on page 1373. The example uses DRAWOVAL to highlight a student’s data point. It draws an oval around the marker symbol that represents the student’s height and weight, and it displays the student’s name inside the oval. In the BEGINGRAPH statement, the setting for the DRAWSPACE= option sets the drawing space and drawing units for the DRAWOVAL and DRAWTEXT statements. In the DRAWOVAL statement, setting TRANSPARENCY=0.75 ensures that the marker for Alfred is visible behind the oval. The DRAWTEXT statement draws the text that identifies the student’s name, using the ANCHOR=, X=, and Y= options to position the text within the oval.
Example Program

```sas
proc template;
    define statgraph drawoval;
        begingraph / drawspace=datavalue;
            entrytitle "Regression Fit Plot";
            layout overlay;
                modelband "myclm";
                scatterplot x=height y=weight;
                regressionplot x=height y=weight / alpha=0.01 clm="myclm";
                drawoval x=69 y=112.5 width=15 height=20 /
                    display=all fillattrs=(color=green)
                        transparency=0.75
                ;
                drawtext "Alfred" / x=69 y=112 anchor=top;
            endlayout;
        endgraph;
    end;

    proc sgrender data=sashelp.class template=drawoval;
        run;
```

DRAWSHADOW Point

Draws a shadow at the point of a marker. Use this statement to draw a shadow that is cast by a marker symbol at a specific location. DRAWSHADOW is similar to SHADECURVE or SHADEPOLYGON except that it is more suitable for certain types of data points. It can be used with both bar and marker charts.
Syntax

**DRAWRECTANGLE**  
\[ X=\text{constant} \mid \text{scalar-expression} \]  
\[ Y=\text{constant} \mid \text{scalar-expression} \]  
\[ \text{WIDTH}=\text{constant} \mid \text{scalar-expression} \]  
\[ \text{HEIGHT}=\text{constant} \mid \text{scalar-expression} \]  
\[ \langle /\text{options} \rangle ; \]

Summary of Optional Arguments

**Appearance options**

*CORNERRADIUS=number*

specifies the radius of the rectangle corners.

*DISPLAY=STANDARD | ALL | (display-options)*

specifies the features to display for the rectangle.

*FILLATTRS=style-element | style-element (fill-options) | (fill-options)*

specifies the fill attributes for this draw statement.

*LAYERVERSE_EQUALITY=FRONT | BACK*

specifies the layer on which this draw statement's output is drawn.

*OUTLINEATTRS=style-element | style-element (line-options) | (line-options)*

specifies the outline attributes for this draw statement.

*ROTADE=number*

specifies the angle of rotation for the rectangle, measured in degrees.

*TRANSPARENCY=number*

specifies the degree of the transparency of this draw statement's output.

**Axes options**

*DISCRETEOFFSET=number*

specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

*XAXIS=X | X2*

specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

*YAXIS=Y | Y2*

specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

**Drawing space options**

*DRAWSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE*

specifies a default drawing space and drawing units for this draw statement.

*XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE*

specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

*YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE*
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Location options

ANCHOR= CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT

specifies an anchor point for the rectangle.

ODS options

URL="string" | string-expression

specifies an HTML page that is displayed when the output of this draw statement is selected.

Size options

HEIGHTUNIT= PERCENT | PIXEL | DATA

specifies whether the positive-number that is specified in the HEIGHT= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

WIDTHUNIT= PERCENT | PIXEL | DATA

specifies whether the positive-number that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Required Arguments

X=constant | scalar-expression

specifies the X value of the anchor point.

Interactions

The value that is set for this argument is interpreted using the XSPACE= option.

When XSPACE=DATAVALUE, the value is interpreted using the XAXIS= option.

Note

When XSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

Y=constant | scalar-expression

specifies the Y value of the anchor point.

Interactions

The value that is set for this argument is interpreted using the YSPACE= option.

When YSPACE=DATAVALUE, the value is interpreted using the YAXIS= option.

Note

When YSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

WIDTH=constant | scalar-expression

specifies the width of the rectangle.

Interactions

The value that is set for this argument is interpreted using the WIDTHUNIT= and XSPACE= options.
When WIDTHUNIT=DATA, the value is interpreted using the XAXIS= option.

**HEIGHT=constant | scalar-expression**  
specifies the height of the rectangle.

Interactions  
The value that is set for this argument is interpreted using the HEIGHTUNIT= and YSPACE= options.

When HEIGHTUNIT=DATA, the value is interpreted using the YAXIS= option.

### Optional Arguments

**ANCHOR=** CENTER | TOPLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT  
specifies an anchor point for the rectangle. The anchor point is relative to the unrotated rectangle. The anchor point can be at the center of the rectangle or at eight points on the rectangle border. The following figure shows the anchor points for TOPLEFT and CENTER.

![Anchor Points](image)

The coordinates of the anchor point are set by the statement’s X and Y values, and by the settings for the XSPACE= and YSPACE= options. The XAXIS= and YAXIS= option might affect positioning when the XSPACE= or YSPACE= options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

When the rectangle is rotated, the anchor point remains relative to the unrotated rectangle while the rectangle is rotated on its anchor point. See **ROTATE=** on page 1378.

Default  CENTER

**CORNERRADIUS=number**  
specifies the radius of the rectangle corners.

Default  0  
Range  0–1, where 0 specifies square corners and 1 specifies the most rounded corners

Example  cornerradius=0.2

**DISCRETEOFFSET=number**  
specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

Default  0 (no offset, output is centered on the discrete X values, or the discrete Y values, or both)

Range  −0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the
right. If the Y axis is discrete, then a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed.

**Restriction**

This option applies only when the options XSPACE= or YSPACE= use DATAVALUE, and when X or Y are values on a discrete axis. For nondiscrete axes, this option is ignored.

**Tip**

Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset.

**DISPLAY=STANDARD | ALL | (display-options)**

specifies the features to display for the rectangle.

**STANDARD**

displays an outlined rectangle.

**ALL**

displays an outlined, filled rectangle.

**(display-options)**

specifies a space-separated list of one or more display options, enclosed in parentheses.

The list can contain one or both of the following options:

- **OUTLINE**
  displays an outlined rectangle

- **FILL**
  displays a filled rectangle

**Default** STANDARD

**Tip**

Use the OUTLINEATTRS= and FILLATTRS= options to control the appearance of the rectangle.

**DRAWSPACE=GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**

specifies a default drawing space and drawing units for this draw statement.

**Default** LAYOUTPERCENT

**Interactions**

This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this draw statement.

This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT=, or WIDTHUNIT=.

**See**

“About the Drawing Space and Drawing Units” on page 1322

**FILLATTRS=style-element | style-element (fill-options) | (fill-options)**

specifies the fill attributes for this draw statement.
### The GraphAnnoShape style element Color attribute

<table>
<thead>
<tr>
<th>Default</th>
<th>The GraphAnnoShape style element Color attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tip</td>
<td>The <code>TRANSPARENCY=</code> option sets the transparency for the fill and the outline. You can combine this option with <code>TRANSPARENCY=</code> to set one transparency for the outline but a different transparency for the fill. Example: <code>transparency=0.2 fillattrs=(transparency=0.6)</code></td>
</tr>
<tr>
<td>See</td>
<td>“General Syntax for Attribute Options” on page 1519 for the syntax on using a <code>style-element</code></td>
</tr>
<tr>
<td></td>
<td>“Fill Color Options” on page 1520 for available <code>fill-options</code>.</td>
</tr>
</tbody>
</table>

### HEIGHTUNIT=PERCENT | PIXEL | DATA

specifies whether the positive-number that is specified in the `HEIGHT=` option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

<table>
<thead>
<tr>
<th>Default</th>
<th>PERCENT</th>
</tr>
</thead>
</table>

**Interaction**

This setting combines with the `YSPACE=` setting to interpret the height that is set in the `HEIGHT=` option.

### LAYER=FRONT | BACK

specifies the layer on which this draw statement's output is drawn.

- FRONT: draws the output on top of the graph.
- BACK: draws the output behind the background areas, such as a layout or legend background.

<table>
<thead>
<tr>
<th>Default</th>
<th>FRONT</th>
</tr>
</thead>
</table>

**Tip**

For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 1325.

### OUTLINEATTRS=style-element | style-element (line-options) | (line-options)

specifies the outline attributes for this draw statement.

<table>
<thead>
<tr>
<th>Default</th>
<th>The GraphAnnoLine style element</th>
</tr>
</thead>
</table>

**Interaction**

For this option to have any effect, the outline must be enabled by the `DISPLAY=` option or by the ODS style that is in effect.

**See**

“General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.

“Line Options” on page 1522 for available `line-options`.

### ROTATE=number

specifies the angle of rotation for the rectangle, measured in degrees. The angle is measured as if a horizontal line extended to the right through the rectangle anchor point as shown in the following figure.
Positive angles rotate the rectangle counter clockwise, and negative angles rotate the rectangle clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0. No rotation is performed

See ANCHOR= on page 1376

**TRANSPARENCY=number**

specifies the degree of the transparency of this draw statement’s output.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

Tip The FILLATTRS= option can be used to set transparency for just the filled area. You can combine this option with FILLATTRS= to set one transparency for the outline but a different transparency for the fill. Here is an example:

```
transparency=0.2 fillattrs=(transparency=0.6)
```

**URL="string" | string-expression**

specifies an HTML page that is displayed when the output of this draw statement is selected.

Note: This option is valid starting in SAS 9.4M1.

"string"

**string-expression**

specifies a valid HTML page reference (HREF) for the graphical element that is drawn by this draw statement.

Example http://www.sas.com

Requirement To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Tip By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to
your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

**WIDTHUNIT=PERCENT | PIXEL | DATA**

specifies whether the positive-number that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

Default PERCENT

Interaction This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= option.

**XAXIS=X | X2**

specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.

Default X

Interaction This option has effect only if XSPACE=DATAVALUE.

**XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**

specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.

Default The setting that is in effect for the DRAWSPACE= option.

**YAXIS=Y | Y2**

specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default Y

Interaction This option has effect only if YSPACE=DATAVALUE.

**YSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE**

specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default The setting that is in effect for the DRAWSPACE= option.

### Details

A DRAWRECTANGLE statement draws a rectangle in a graph. The rectangle position is determined by the X and Y anchor points, and the size is determined by the HEIGHT and WIDTH settings. You can manage the rectangle position with the options ANCHOR=, XSPACE=, and YSPACE=. You can manage the rectangle size with the HEIGHTUNIT= and WIDTHUNIT= options.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, “Key Concepts for Using Draw
Example: DRAWRCTANGLE Statement

The following graph was generated by the "Example Program" on page 1381. The example uses DRAWRCTANGLE to highlight a student's data point. It draws a rectangle around the marker symbol that represents the student's height and weight, and it displays the student's name inside the rectangle. In the BEGINGRAPH statement, the setting for the DRAWSPACE= option sets the drawing space and drawing units for the DRAWRCTANGLE and DRAWTXT statements. In the DRAWRCTANGLE statement, setting TRANSPARENCY=0.75 ensures that the marker for Alfred is visible behind the rectangle. The DRAWTXT statement draws the text that identifies the student's name, using the ANCHOR=, X=, and Y= options to position the text within the rectangle.

Example Program

```
proc template;
  define statgraph drawrectangle;
    begingraph / drawspace=datavalue;
      entrytitle "Regression Fit Plot";
      layout overlay;
        modelband "myclm"
        scatterplot x=height y=weight;
        regressionplot x=height y=weight / alpha=0.01 clm="myclm";
        ddrawrectangle x=69 y=112.5 width=10 height=15 /
          display=all fillattrs=(color=green)
          transparency=0.75 ;
        ddrawtext "Alfred" / x=69 y=112 anchor=top;
  end GRAPH;
end;
```
proc sgrender data=sashelp.class template=drawrectangle;
run;

DRAWTEXT Statement

Draws and anchors in a graph a text box that contains one or more lines of formatted text.

Syntax

```
DRAWTEXT text-item <text-item ...> <options>;
```

Summary of Optional Arguments

**Appearance options**

- `BORDER=TRUE | FALSE` specifies whether a border is drawn around the text box.
- `BORDERATTRS=style-element | style-element (line-options) | (line-options)` specifies the border line attributes for this draw statement.
- `LAYER=FRONT | BACK` specifies the layer on which this draw statement’s output is drawn.
- `PAD=dimension | (pad-options)` specifies the amount of extra space that is reserved inside the text box’s border.
- `ROTATE=number` specifies the angle of rotation of the text box, measured in degrees.
- `TRANSPARENCY=number` specifies the degree of the transparency of this draw statement’s output.

**Axes options**

- `DISCRETEOFFSET=number` specifies an amount to offset from the discrete X values, or the discrete Y values, or both.
- `XAXIS=X | X2` specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.
- `YAXIS=Y | Y2` specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

**Drawing space options**

- `DRAWSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE`
specifies a default drawing space and drawing units for this draw statement.

\[ \text{XSPACE} = \text{GRAPHPERCENT} \mid \text{GRAPHPIXEL} \mid \text{LAYOUTPERCENT} \mid \text{LAYOUTPIXEL} \mid \text{WALLPERCENT} \mid \text{WALLPIXEL} \mid \text{DATAPERCENT} \mid \text{DATAPIXEL} \mid \text{DATAVALUE} \]

specifies the drawing space and drawing units for interpreting the value that is specified in the \text{X=} option.

\[ \text{YSPACE} = \text{GRAPHPERCENT} \mid \text{GRAPHPIXEL} \mid \text{LAYOUTPERCENT} \mid \text{LAYOUTPIXEL} \mid \text{WALLPERCENT} \mid \text{WALLPIXEL} \mid \text{DATAPERCENT} \mid \text{DATAPIXEL} \mid \text{DATAVALUE} \]

specifies the drawing space and drawing units for interpreting the value that is specified in the \text{Y=} option.

**Location options**

\[ \text{ANCHOR} = \text{CENTER} \mid \text{TOPLEFT} \mid \text{TOP} \mid \text{TOPRIGHT} \mid \text{LEFT} \mid \text{RIGHT} \mid \text{BOTTOMLEFT} \mid \text{BOTTOM} \mid \text{BOTTOMRIGHT} \]

specifies an anchor point for the text box.

\[ \text{JUSTIFY} = \text{LEFT} \mid \text{CENTER} \mid \text{RIGHT} \]

specifies the alignment of text that wraps within the text box.

\[ \text{X=} \text{constant} \mid \text{scalar-expression} \]

specifies the anchor point’s X coordinate.

\[ \text{Y=} \text{constant} \mid \text{scalar-expression} \]

specifies the anchor point’s Y coordinate.

**ODS options**

\[ \text{URL} = "\text{string}" \mid \text{string-expression} \]

specifies an HTML page that is displayed when the output of this draw statement is selected.

**Size options**

\[ \text{WIDTH=} \text{positive-number} \]

specifies the width of the text box.

\[ \text{WIDTHUNIT=} \text{PERCENT} \mid \text{PIXEL} \mid \text{DATA} \]

specifies whether the \text{positive-number} that is specified in the \text{WIDTH=} option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.

**Required Argument**

\[ \text{text-item} \ <\ldots\text{text-item}> \]

specifies one or more pieces of text for the text box. Each \text{text-item} has the following form:

\[ <\text{prefix-option}> "\text{string}" \mid \text{dynamic} \mid \text{character-expression} \mid \{\text{text-command}\} \]

Each piece of text can have a prefix setting that precedes the piece of text. A piece of text is either a string literal, a dynamic, or a text command. All \text{text-items} are concatenated into one string, which might be wrapped, based on the settings for the \text{WIDTH=} and \text{WIDTHUNIT=} settings. Leading and trailing blanks in the concatenated string are always used.

When used, a prefix option applies to the immediately following piece of text and also to all subsequent text strings and text-commands until another prefix option is specified.
Requirements

string must be enclosed in quotation marks.

character-expression must be enclosed in an EVAL function.

text-command must be enclosed in braces.

Note

Leading and trailing spaces are preserved.

See

Chapter 10, “Managing Text Items,” on page 1265

Optional Arguments

ANCHOR= CENTER | TOLEFT | TOP | TOPRIGHT | LEFT | RIGHT | BOTTOMLEFT | BOTTOM | BOTTOMRIGHT

specifies an anchor point for the text box. The anchor point is relative to the unrotated text. It can be at the center of the text box or at eight points on the border of the text box bounding box. The following figure shows the anchor points for TOLEFT and LEFT.

The coordinates of the anchor point are set by the \texttt{X=} and \texttt{Y=} options, and by the \texttt{XSPACE=} and \texttt{YSPACE=} options. The \texttt{XAXIS=} and \texttt{YAXIS=} option might affect positioning when the \texttt{XSPACE=} or \texttt{YSPACE=} options are set to DATAPIXEL, DATAPERCENT, or DATAVALUE.

The text box has a fixed width, determined by the \texttt{WIDTH=} and \texttt{WIDTHUNIT=} options. The height of the text box is based on the amount of text specified and the font size. The height of the text grows in a direction that is related to the anchor point. For example, if \texttt{ANCHOR=TOLEFT}, then the text box height extends downward from the anchor point and its width extends to the right. If \texttt{ANCHOR=CENTER}, then half the text box width and half the text box height extend equally left and right, as well as top to bottom from the anchor point. If \texttt{ANCHOR=BOTTOM}, the text box height extends upward from the anchor point and the text box width is centered at the anchor point.

When the text is rotated, the anchor point remains relative to the unrotated text box while the text box is rotated on its anchor point. See \texttt{ROTATE=} on page 1387.

Default CENTER

BORDER= TRUE | FALSE

specifies whether a border is drawn around the text box.

Default FALSE

Tip Use the \texttt{BORDERATTRS=} option to control the appearance of the border.

See “boolean” on page 1479 for other Boolean values that you can use.
**BORDERATTRS=**`style-element | style-element (line-options) | (line-options)`  
specifies the border line attributes for this draw statement.

<table>
<thead>
<tr>
<th>Default</th>
<th>The GraphBorderLines style element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction</td>
<td>BORDER=TRUE must be set for this option to have any effect.</td>
</tr>
</tbody>
</table>
| See           | “General Syntax for Attribute Options” on page 1519 for the syntax on using a `style-element`.  
                “Line Options” on page 1522 for available `line-options`. |

**DISCRETEOFFSET=**`number`  
specifies an amount to offset from the discrete X values, or the discrete Y values, or both.

| Default | 0 (no offset, output is centered on the discrete X values, or the discrete Y values, or both) |
| Range   | –0.5 to +0.5, where 0.5 represents half the distance between discrete ticks. If the X axis is discrete, then a positive offset is to the right. If the Y axis is discrete, then a positive offset is up. If REVERSE=TRUE on the X or Y axis, then the offset direction is also reversed. |
| Restriction | This option applies only when the options XSPACE= or YSPACE= use DATAVALE, and when X or Y are values on a discrete axis. For nondiscrete axes, this option is ignored. |
| Tip     | Setting the discrete offset for the plots does not affect the axis minimum and maximum offsets. In some cases, setting a discrete offset can cause clipping at each end of the axis. In those cases, use the OFFSETMIN= and OFFSETMAX= axis options to increase the axis minimum and maximum offsets to accommodate the discrete offset. |

**DRAWSPACE=**`GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALE`  
specifies a default drawing space and drawing units for this draw statement.

| Default | LAYOUTPERCENT |
| Interactions | This statement and all of the draw statements inherit the global DRAWSPACE= setting from the DRAWSPACE= option in the BEGINGRAPH statement. Setting this option changes the setting for only this draw statement.  
                This option sets the default drawing space, but individual settings in the X or Y dimension can be overridden by the options XSPACE=, YSPACE=, HEIGHTUNIT=, or WIDTHUNIT=. |
| See     | “About the Drawing Space and Drawing Units” on page 1322 |

**JUSTIFY=**`LEFT | CENTER | RIGHT`  
specifies the alignment of text that wraps within the text box.
LEFT
forces the first character of each line to appear at the left margin (distance from the left border plus the left pad amount).

CENTER
forces each line to be centered in the text box between the left and right pad amounts.

RIGHT
forces the last character of each line to appear at the right margin (distance from the right border minus the right pad amount).

Default LEFT

Interaction Text is wrapped based on the width of the specified text, the font size, and the setting in the PAD= option.

LAYER=FRONT | BACK
specifies the layer on which this draw statement’s output is drawn.

FRONT
draws the output on top of the graph.

BACK
draws the output behind the background areas, such as a layout or legend background.

Default FRONT

Tip For elements that are obstructed because they are in the back layer, you can suppress the display of filled areas in the graph. You can also use transparency to manage the element visibility. For more information, see “About Drawing Layers” on page 1325.

PAD=dimension | (pad-options)
specifies the amount of extra space that is reserved inside the text box’s border.

\textit{dimension}
specifies a dimension to use for the extra space at the left, right, top, and bottom of the text box border.

\textit{(pad-options)}
a space-separated list of one or more name-value-pair pad options enclosed in parentheses.

The list can include one or more of the following pad options:

LEFT=dimension
specifies the amount of extra space added to the left side.

RIGHT=dimension
specifies the amount of extra space added to the right side.

TOP=dimension
specifies the amount of extra space added to the top.

BOTTOM=dimension
specifies the amount of extra space added to the bottom.

Note Sides that are not assigned padding are padded with the default amount.
Tip Use `pad-options` to create non-uniform padding.

Default (LEFT=3 RIGHT=3 TOP=0 BOTTOM=0)

Note The default units for `dimension` are pixels.

See “`dimension`” on page 1480

**ROTATE=number**

specifies the angle of rotation of the text box, measured in degrees. The text box is rotated around its anchor point. The angle is measured from a horizontal line passing through the anchor point of the text box to the right. The following figure shows the rotation of text around a top anchor point.

Positive angles rotate the text box counter clockwise, and negative angles rotate the text box clockwise. The angle specification can exceed 360 degrees in absolute value.

Default 0. No rotation is performed

See `ANCHOR=` on page 1384

**TRANSPARENCY=number**

specifies the degree of the transparency of this draw statement’s output.

Default 0

Range 0–1, where 0 is opaque and 1 is entirely transparent

**URL=”string” | string-expression**

specifies an HTML page that is displayed when the output of this draw statement is selected.

Note: This option is valid starting in SAS 9.4M1.

```
“string”
```

`string-expression`

specifies a valid HTML page reference (HREF) for the graphical element that is drawn by this draw statement.

Example http://www.sas.com
Requirement  To generate selectable graphical elements, you must include an ODS GRAPHICS ON statement that specifies the IMAGEMAP option, and you must write the output to the ODS HTML destination.

Tip  By default, drill-down links open in a new browser window (link target _blank). To specify a different target for your drill-down links, such as _self or _parent, add the DRILLTARGET="target" option to your ODS GRAPHICS statement. See DRILLTARGET= on page 1489.

WIDTH=positive-number
specifies the width of the text box.
Default  10
Interaction  This option's value is interpreted using the WIDTHUNIT= option.

WIDTHUNIT=PERCENT | PIXEL | DATA
specifies whether the positive-number that is specified in the WIDTH= option is interpreted as a percentage value, a pixel value, or a value that is in the units of the data.
Default  PERCENT
Interaction  This setting combines with the XSPACE= setting to interpret the width that is set in the WIDTH= option.

X=constant | scalar-expression
specifies the anchor point's X coordinate.
Default  50
Interactions  The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the XSPACE= option.

If XSPACE=DATAVALUE, then this option's value is interpreted using the XAXIS= option.

Note  When XSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

XAXIS=X | X2
specifies whether the data value for the arguments X1 and X2 is interpreted using the primary X-axis scale or to the secondary X (X2) axis scale.
Default  X
Interaction  This option has effect only if XSPACE=DATAVALUE.

XSPACE=GRAPHPERCENT | GRAPHPIXEL | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPIXEL | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified in the X= option.
The setting that is in effect for the DRAWSPACE= option.

Y=constant | scalar-expression
specifies the anchor point’s Y coordinate.

Default 50

Interactions The DRAWSPACE= option determines the default interpretation of the units for this setting. You can override the default with the YSPACE= option.

If YSPACE=DATAVALUE, then this option’s value is interpreted using the YAXIS= option.

Note When YSPACE=DATAVALUE and a character value is specified, leading blanks are honored and trailing blanks are ignored when the specified value is compared with the data values.

YAXIS=Y | Y2
specifies whether the data value for the arguments Y1 and Y2 is interpreted using the primary Y-axis scale or to the secondary Y (Y2) axis scale.

Default Y

Interaction This option has effect only if YSPACE=DATAVALUE.

YSPACE=GRAPHPERCENT | GRAPHPixel | LAYOUTPERCENT | LAYOUTPIXEL | WALLPERCENT | WALLPIXEL | DATAPERCENT | DATAPixel | DATAVALUE
specifies the drawing space and drawing units for interpreting the value that is specified in the Y= option.

Default The setting that is in effect for the DRAWSPACE= option.

Prefix Option

TEXTATTRS=style-element | style-element (text-options) | (text-options)
specifies the color and font properties of the entire text string or individual text-items.

Default The GraphAnnoText style element.

Interaction When multiple TEXTATTRS= prefix options are used, each one cancels the last, resetting all text properties to the default set by the GraphAnnoText style element. Subsequent text-items to the right are then assigned the text properties specified in the closest TEXTATTRS= setting to their left. Thus, to vary the text properties across text-items, you do not have to override settings from a previous TEXTATTRS= setting. Each TEXTATTRS= specification resets all text properties to the default so that only the new settings are applied to subsequent text-items.

Tip To ensure that all text has the same text properties, use this prefix option once only and place it before the first text-item.

See “General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.
Text Commands

{ SUB "string" | dynamic }

text-command that specifies that the string or dynamic is to appear as subscript text.

See "Rules for Unicode and Special Character Specifications" on page 1270

Example drawtext "y = " b{sub "0"} " + b{sub "1"} "x";

{ SUP "string" | dynamic }

text-command that specifies that the string or dynamic is to appear as superscript text.

See "Rules for Unicode and Special Character Specifications" on page 1270

Example drawtext "R" {sup "2"} " = " {format (6.4) RSQUARED} ;

{ UNICODE "hex-string"x | keyword | dynamic }

text-command that specifies a glyph (character) to be displayed using its Unicode specification or keyword equivalent.

"hex-string"x

a four-byte hexadecimal constant that represents a UNICODE character in the current font. For a complete listing, see http://unicode.org/charts/charindex.html.

keyword

a SAS keyword for a UNICODE character.

For a listing of SAS supplied keywords, see Appendix 3, “Reserved Keywords and Unicode Values,” on page 1515.

dynamic

a dynamic variable that resolves to either "hex-string"x or a keyword for a UNICODE character.

Note The UNICODE text command attempts to access a UNICODE value in the current font. Not all fonts support accessing characters through their UNICODE value. Some fonts support only a limited set of UNICODE values. If the UNICODE value is not accessible, then the command might be ignored or a nonprintable character might be substituted.

See "Using UNICODE Text" on page 1269

"Rules for Unicode and Special Character Specifications" on page 1270

Example The following statements show how to use the {UNICODE} text command:

drawtext {unicode alpha} "=" CONF;

drawtext {unicode "03B1"x} "=" CONF;
Details

A DRAWTEXT statement draws a text box that contains one or more lines of text. The text can be formatted, using the TEXTATTRS= prefix option. By default, the text box is drawn in the center of the graph. You can change the default position with the options ANCHOR=, X=, Y=, XSPACE=, and YSPACE=.

For general information about the types of elements that can be drawn with the draw statements, the drawing space and drawing units that they use, and how the drawn elements are anchored, see Chapter 13, "Key Concepts for Using Draw Statements," on page 1319. For detailed usage information, consult the SAS Graph Template Language: User’s Guide.

Example: DRAWTEXT Statement

The following graph was generated by the "Example Program" on page 1391. The first DRAWTEXT statement shows how to draw multiple lines of text in a specific position within the graph. The second DRAWTEXT statement shows how to create a watermark, which is achieved by applying transparency to text that is rotated within the graph.

Example Program

```sas
proc template;
define statgraph modelfit;
begingraph;
  entrytitle "Regression Fit Plot";
  layout lattice;
  layout overlay / xaxisopts=(offsetmin=0.1);
  drawtext textattrs=(style=italic size=8pt)
    "Band shows 99% Confidence Limit of Mean" /
```

CONFIDENTIAL
anchor=bottomleft width=15 widthunit=percent xspace=wallpercent yspace=wallpercent x=0 y=10 justify=center;
modelband "myclm";
scatterplot x=height y=weight / primary=true;
regressionplot x=height y=weight / alpha=0.01 clm="myclm";
endlayout;
endlayout;
drawtext textattrs=(color=gray size=52pt) "CONFIDENTIAL" /
transparency=0.75 rotate=-35
width=110 widthunit=percent justify=center;
endgraph;
end;

proc sgrender data=sashelp.class template=modelfit;
run;
PART 10

GTL Annotation Facility

Chapter 15

About the GTL Annotation Facility ........................................ 1395

Chapter 16

The ANNOTATE Statement .................................................. 1403
Data-Set-Driven Annotations

GTL supports data-set-driven annotations, which enable you to add the following graphics elements to your graphs:

- text
- lines and arrows
- circles and ovals
- squares and rectangles
- polygons and polylines
- images

Unlike graphics elements that are drawn using GTL draw statements in a GTL template, data-set-driven annotations are drawn from graphics instructions that are stored in a SAS data set. The GTL annotation facility is similar to the SAS/GRAPH Annotate facility. The GTL annotation facility enables you to separate your annotation instructions from your template statements. To change your annotations, you can specify a different annotation data set or modify the instructions in the original data set. You do not have to modify your template code.

The following items are required in order to use the GTL annotation facility:

- a SAS data set that contains the annotation instructions. The annotation instructions must be stored in a SAS data set. Annotation instructions stored in a Cloud Analytic Services (CAS) in-memory table are not supported.
- at least one ANNOTATE statement in your GTL template.
Annotation Data Set

About the Annotation Data Set

The annotation data set contains the instructions for drawing annotations. Each observation in the data set contains columns for an annotation instruction. The following information is typically stored in these columns:

- the annotation function name
- the coordinates of the annotation location
- the drawing space
- the attributes of the annotation, such as color, font, and so on

The information that is required for each instruction is function-dependent. See "Annotation Data Requirements" on page 1396.

To create the annotation data set, use the same methods that you would use to create any SAS data set. You must store the annotation instructions in a SAS data set. Annotation instructions stored in a CAS in-memory table are not supported. The most common method for creating a data set is to use a DATA step. For more information about the DATA step, see SAS Language Reference: Concepts. Macros are available that you can run in your DATA step to create the observations for your annotations. See “Using the SGANNO Macros” on page 1399.

Annotation Data Requirements

Each observation in the annotation data set must provide sufficient information to complete an annotation instruction. To provide this information, certain columns must be included in each observation. All instructions require a Function column, which is a character column that stores the name of the function that is to be performed. An annotation function draws an annotation, such as a line, arrow, oval, and so on. The Function column can specify one of the following function names:

<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARROW</td>
<td>POLYGON</td>
<td>TEXT</td>
</tr>
<tr>
<td>IMAGE</td>
<td>POLYLINE</td>
<td>TEXTCONT</td>
</tr>
<tr>
<td>LINE</td>
<td>POLYCONT</td>
<td></td>
</tr>
<tr>
<td>OVAL</td>
<td>RECTANGLE</td>
<td></td>
</tr>
</tbody>
</table>
For more information about these functions, see “SG Annotation Function Dictionary” in SAS ODS Graphics: Procedures Guide.

Each function requires one or more additional columns that provide required information such as the location coordinates, dimensions, and so on. The following table lists the additional columns that are required for each function. At a minimum, you must include the Function column and the columns listed in the following table in each observation in your annotation data set.

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>Additional Columns Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARROW</td>
<td>X1 or XC1</td>
<td>These columns store the starting coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 or XC2</td>
<td>These columns store the ending coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y2 or YC2</td>
<td></td>
</tr>
<tr>
<td>IMAGE</td>
<td>Image</td>
<td>This column stores the path to the image file (character).</td>
</tr>
<tr>
<td>LINE</td>
<td>X1 or XC1</td>
<td>These columns store the starting coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X2 or XC2</td>
<td>These columns store the ending coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y2 or YC2</td>
<td></td>
</tr>
<tr>
<td>OVAL</td>
<td>Height</td>
<td>These columns store the oval dimensions (numeric).</td>
</tr>
<tr>
<td></td>
<td>Width</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X1 or XC1</td>
<td>These columns store the oval anchor coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
<tr>
<td>POLYGON</td>
<td>X1 or XC1</td>
<td>These columns store the polygon starting coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
<tr>
<td>POLYLINE</td>
<td>X1 or XC1</td>
<td>These columns store the polyline starting coordinates (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
<tr>
<td>POLYCONT</td>
<td>X1 or XC1</td>
<td>These columns store the ending coordinates of a polygon or polyline segment (numeric or character).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
<tr>
<td>RECTANGLE</td>
<td>Height</td>
<td>These columns store the rectangle dimensions (numeric).</td>
</tr>
<tr>
<td></td>
<td>Width</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X1 or XC1</td>
<td>These columns store the rectangle anchor coordinates (numeric).</td>
</tr>
<tr>
<td></td>
<td>Y1 or YC1</td>
<td></td>
</tr>
</tbody>
</table>
### About the Coordinate Columns

The annotation coordinates specify the location of the annotation as X and Y values. In some cases, such as for a line or arrow, two sets of coordinates are required in order to specify the beginning and ending locations of the annotation. You can specify numeric or character columns for each coordinate value. For numeric coordinate values, the location or starting location coordinates are stored in the X1 and Y1 columns in the annotation data set. When ending coordinates are required, the numeric coordinates are stored in the X2 and Y2 columns. For character values, the coordinates are stored in the XC1 and YC1 columns, and when required, the XC2 and YC2 columns.

Note: If both the numeric and character columns are specified for a coordinate value, then the numeric column takes precedence and the character column is ignored.

### About the Drawing Space

When you specify X and Y coordinates for an annotation, the coordinate values refer to a drawing space. By default, this space is a percentage of the graph area. Specifying X=50 and Y=50, for example, refers to a point in the center of the graph area. Here is a complete list of the drawing spaces that you can use.

<table>
<thead>
<tr>
<th>DATAPERCENT</th>
<th>GRAPHPERCENT</th>
<th>LAYOUTPIXEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAPIXEL</td>
<td>GRAPHPIXEL</td>
<td>WALLPERCENT</td>
</tr>
<tr>
<td>DATAVALUE</td>
<td>LAYOUTPERCENT</td>
<td>WALLPIXEL</td>
</tr>
</tbody>
</table>
For more information about these drawing spaces, see “About the Drawing Space and Drawing Units” on page 1322.

To change the default drawing space for your annotations, you can add the columns shown in the following table to your annotation observations.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DrawSpace</td>
<td>Specifies the drawing space for all coordinates. This value cannot be used with the POLYCONT function.</td>
</tr>
<tr>
<td>X1Space</td>
<td>Specifies the drawing space for the X1 coordinate. This value cannot be used with the TEXTCONT function.</td>
</tr>
<tr>
<td>X2Space</td>
<td>Specifies the drawing space for the ending X2 coordinate. This value can be used with the ARROW and LINE functions only.</td>
</tr>
<tr>
<td>Y1Space</td>
<td>Specifies the drawing space for the Y1 coordinate. This value cannot be used with the TEXTCONT function.</td>
</tr>
<tr>
<td>Y2Space</td>
<td>Specifies the drawing space for the ending Y2 coordinate. This value can be used with the ARROW and LINE functions only.</td>
</tr>
</tbody>
</table>

For information about these columns, see “SG Annotation Function Dictionary” in SAS ODS Graphics: Procedures Guide

Using the SGANNO Macros

Table 15.1  SG Annotation Macros

<table>
<thead>
<tr>
<th>Macro Name¹</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%SGANNO</td>
<td>Compiles the available macros and makes them available for you to use. Note: You must run the %SGANNO macro in your SAS session before you can use any of the other annotation macros that are listed in this table.</td>
</tr>
<tr>
<td>%SGANNO_HELP</td>
<td>Displays help information for a specified annotation macro.</td>
</tr>
<tr>
<td>%SGARROW</td>
<td>Creates an observation that draws an arrow.</td>
</tr>
<tr>
<td>%SGIMAGE</td>
<td>Creates an observation that displays an image.</td>
</tr>
<tr>
<td>%SGLINE</td>
<td>Creates an observation that draws a line.</td>
</tr>
<tr>
<td>%SGOVAL</td>
<td>Creates an observation that draws an oval.</td>
</tr>
<tr>
<td>Macro Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>%SGPOLYCONT</td>
<td>Creates an observation that draws a polygon or polyline segment.</td>
</tr>
<tr>
<td>%SGPOLYGON</td>
<td>Creates an observation that specifies the starting point of a polygon.</td>
</tr>
<tr>
<td></td>
<td>Note: Use the %SGPOLYCONT macro to draw the segments.</td>
</tr>
<tr>
<td>%SGPOLYLINE</td>
<td>Creates an observation that specifies the starting point of a polyline.</td>
</tr>
<tr>
<td></td>
<td>Note: Use the %SGPOLYCONT macro to draw the segments.</td>
</tr>
<tr>
<td>%SGRECTANGLE</td>
<td>Creates an observation that draws a rectangle.</td>
</tr>
<tr>
<td>%SGTEXT</td>
<td>Creates an observation that draws a single line of text or the first line of text in a multiline annotation.</td>
</tr>
<tr>
<td></td>
<td>Note: For multiple lines of text, use the %SGTEXTCONT macro for the subsequent lines.</td>
</tr>
<tr>
<td>%SGTEXTCONT</td>
<td>Creates an observation that draws a continuing line of text in a multiline annotation.</td>
</tr>
</tbody>
</table>

1 The macros listed in this table are valid starting with SAS 9.4M1. You can run these macros in a DATA step to simplify the process of creating your SG annotation data sets.

For more information about these macros, see SAS ODS Graphics: Procedures Guide.

See Also

“Adding Observations for Polylines and Polygons” in SAS Graph Template Language: User’s Guide

ANNOTATE Statement

To render the annotations in your annotation data set, you must include at least one ANNOTATE statement in the GTL template for your graph. You can place the ANNOTATE statement anywhere in the template to render your annotations. The annotations are drawn using the context in which the ANNOTATE statement is encountered. You can use the ANNOTATE statement with no options to render all of the annotations in your annotation data set. You can use the ID= option in the ANNOTATE statement to render only a subset of the annotations.
For more information about the ANNOTATE statement, see "ANNOTATE Statement" on page 1403.

SGRENDER Statement SGANNO Option

The SGRENDER statement SGANNO= option specifies the name of a SAS data set that contains annotation instructions. To render your graph with annotations, you must use the SGANNO= option in the SGRENDER statement to specify the name of your annotation data set. When the ANNOTATE statement is encountered in the template, the annotation instructions are read from the annotation data set and are rendered in the current context in the graph. If the template does not include the ANNOTATE statement, the annotations in the annotation data set are ignored. See "SGANNO=annotation-data-set" in SAS ODS Graphics: Procedures Guide.
The ANNOTATE Statement

**Dictionary**

**ANNOTATE Statement**

Draws annotations from annotation instructions that are stored in a SAS data set.

**Requirements:**
- The annotation instructions must be stored in a SAS data set.
- The annotation data set must be specified by the SGANNO= option in the SGRENDER statement.

**Tips:**
- When the ANNOTATE statement is placed inside a LAYOUT container, the layout container is used as the context when DRAWSPACE for an annotation is specified as LAYOUT, WALL, or DATA.
- When the DRAWSPACE for an annotation object is GRAPH, the context for drawing the object is the graph area regardless of where the ANNOTATE statement is located.
- The LAYER option determines whether the annotations are drawn in front of the graph elements or behind them.

**See:**
- “Annotation Data Set” on page 1396
- SGRENDER statement SGANNO= option
- “Adding Data-Driven Annotations to Your Graph” in *SAS Graph Template Language: User’s Guide*

**Syntax**

```
ANNOTATE </ID=“annotation-ID”>
```

**Optional Argument**

**ID=“annotation-identifier”**

specifies the ID column value of the annotations that are to be drawn. The ID column of the annotation data set contains a unique character value that
identifies the subset to which each annotation belongs. All annotations in the annotation data set with an ID column value that matches the specified annotation identifier are drawn. If the annotation data set does not contain an ID column or if no ID column value matches the specified identifier, then no annotations are drawn.

**Default**
All of the annotations in the annotation data set are drawn.

**Requirement**
The annotation data set must contain an ID column, and at least one ID column value must match the specified identifier value.

**Tips**
The ID= option can be used with the annotation data set ID column to subset the annotations that are stored in an annotation data set.

The identifier value matching is case sensitive.

**See**
“Subsetting Annotations” in SAS Graph Template Language: User’s Guide

---

**Details**
You can use the ANNOTATE statement anywhere in a GTL template to render annotation objects in the current context. The annotation objects are read from an annotation data set that is specified by the SGANNO= option in the SGRENDER statement. You must use at least one ANNOTATE statement in the template to draw the annotations. By default, all of the annotations in the annotation data set are drawn. You can use the ID column in the annotation data set with the ID= option in the ANNOTATE statement to draw only a subset of the annotations in the data set, if you want to. See “Subsetting Annotations” in SAS Graph Template Language: User’s Guide.

**Examples:**

**Example 1: Adding Simple Text Annotations**
Here is an example that uses the GTL annotation facility to place custom Y-axis labels inside the plot wall for each of two side-by-side plots. Each label is rotated 90 degrees and is placed on the inside edge of the plot wall near the plot’s Y axis. The default Y-axis labels are suppressed.

Here is the output for this example.
/* Create the annotation data set */
data anno;
  length label $30 id $5 anchor $12;
drawspace="wallpercent";

  /* Create the bar chart Y-axis label. */
id='BAR'; function='text'; x1=0;
textweight='bold'; anchor='top'; rotate=90;
width=1000; widthunit="pixel";
label="MPG (City)";
output;

  /* Create the histogram Y-axis label. */
id='HIST'; anchor='bottom'; x1=100;
label="Distribution of MPG (Percent)";
output;
run;

/* Define the template */
proc template;
define statgraph anno;
begingraph;
  entrytitle "Vehicle Statistics";
  layout lattice / columns=2 columngutter=10;
  layout overlay /
    xaxisopts=(offsetmin=0.2 offsetmax=0.2)
yaxisopts=(display=(ticks tickvalues));

  /* Draw the barchart of origin and MPG city */
barchart x=origin y=mpg_city / name="bar" stat=mean
    group=type groupdisplay=cluster clusterwidth=0.7
dataSkin=sheen;
discretelegend "bar";
Program Description

Create the annotation data set Anno. The annotation data set contains two observations, one for each label. The DRAWSPACE is set to WALLPERCENT for both labels. The ID column specifies an identifier for each label. The bar chart label is identified as BAR. It is rotated 90 degrees, anchored on TOP, and is placed at 0% of the wall space along the X axis. The default value 50% percent is used along the Y axis to center the label vertically. This places the label on inside left edge of the bar chart wall. This histogram label is identified as HIST. It is rotated 90 degrees, anchored on BOTTOM, and is placed at 100% of the wall space along the X axis. Like the bar chart label, the default 50% percent is used along the Y axis. This places the label on the right inside edge of the histogram plot wall.

Define the graph template. The template defines a two-column, one-row lattice for the two plots. For both plots, the DISPLAY= Y-axis option suppresses the default axis label. For the bar chart, the OFFSETMIN= and OFFSETMAX= X-axis options reserve space for the Y-axis label. The ANNOTATE statement draws the axis label. The ID="BAR" option in the ANNOTATE statement draws the bar chart Y-axis label in the context of the bar chart's OVERLAY layout. For the histogram, the
ANNOTATE statement draws the histogram Y-axis label (ID="HIST") in the context of the histogram’s OVERLAY layout.

```
/* Define the template */
proc template;
  define statgraph anno;
  begingraph;
    entrytitle "Vehicle Statistics";
    layout lattice / columns=2 columngutter=10;
    layout overlay /
      xaxisopts=(offsetmin=0.2 offsetmax=0.2)
                yaxisopts=(display=(ticks tickvalues));

    /* Draw the barchart of origin and MPG city */
    barchart x=origin y=mpg_city / name="bar" stat=mean
               group=type groupdisplay=cluster clusterwidth=0.7
               dataskin=sheen;
    discretelegend "bar";
    annotate / id="BAR"; /* Draw the barchart label. */
    endlayout;

    layout overlay /
      y2axisopts=(display=(ticks tickvalues));

    /* Draw the histogram of MPG city */
    histogram mpg_city / dataskin=sheen yaxis=y2;
    densityplot mpg_city / yaxis=y2;
    annotate / id="HIST"; /* Draw the histogram label. */
    endlayout;
  endlayout;
endgraph;
end;
run;
```

**Render the graph with the annotations.** The SGANNO=ANNO option is included in the SGRENDER statement to specify the name of the annotation data set. When the graph is rendered, the ANNOTATE statements in the graph template cause the annotations to be drawn.

```
/* Render the graph with the annotation */
proc sgrender data=sashelp.cars template=anno sganno=anno;
  where type in ("Sedan" "Sports" "SUV");
run;
```

**Example 2: Using the SG Annotation Macros to Generate Your Annotation Data**

In “Example 1: Adding Simple Text Annotations” on page 1404, standard SAS DATA step statements are used to create the observations for the text annotations. This example shows you how to use the SG annotation macros instead to create the same data set.

**Example Program**

```
/* Create the annotation data set */
data anno;
```
Program Description

Here is the SAS code.

Open the DATA step and run the %SGANNO macro. After you open the DATA step, you must run the %SGANNO macro to compile the SG annotation macros and make them available for your use. You need to compile the macros only once during a SAS session.

/* Create the annotation data set */
data anno;
   /* Compile the annotation macros */
   %sganno;
/* Create the observation for the bar chart Y-axis label. Use the %SGTEXT macro to generate the observation for the bar chart Y-axis label. */
   /* Create the bar chart Y-axis label. */
   %sgtext(
      reset=all,
      id=BAR,
      drawspace=wallpercent,
      x1=0,
      label="MPG (City)",
      textweight=bold,
      anchor=top,
      rotate=90,
      width=1000,
      widthunit=pixel);
/* Create the histogram Y-axis label, and close the DATA step. Specify the new values for the ID, X1, Label, and Anchor columns. To carry over the remaining column values from the previous macro call, do not specify RESET=ALL. */
   /* Create the histogram Y-axis label. */
   %sgtext(
      id=HIST,
      x1=100,
      label="Distribution of MPG (Percent)",
      anchor=bottom);
run;

Chapter 16 / The ANNOTATE Statement
Example 2: Using the SG Annotation Macros to Generate Your Annotation Data

```sas
%sgtext(
    id=HIST,
    x1=100,
    label="Distribution of MPG (Percent)",
    anchor=bottom);
run;
```

See Also

- Chapter 15, “About the GTL Annotation Facility,” on page 1395
- “Adding Data-Driven Annotations to Your Graph” in SAS Graph Template Language: User’s Guide
PART 11

Attribute Maps

Chapter 17
Key Concepts for Using Attribute Maps ........................................... 1413

Chapter 18
Discrete Attribute Map Statements .................................................. 1425

Chapter 19
Range Attribute Map Statements ..................................................... 1439
Key Concepts for Using Attribute Maps

About Attribute Maps

By default, many of the graphical attributes of a plot vary with the plot data. For example, when plots display grouped values, by default, the graphical attributes for each group value are selected from the GraphData1–GraphDataN style elements in data order. Changes in the data order can significantly change the appearance of the plot. When plots display a color gradient, by default, the colors assigned to the classification variable are derived from a color ramp based on the actual range of the data. The color assigned to each value can vary with the range of the classification values. Attribute maps enable you to assign the same graphical properties to specific values or ranges of values regardless of data order or the data range. They are useful when you want your graphs to have consistent visual properties when the data varies.

GTL supports two types of attribute maps: discrete attribute maps and range attribute maps. A discrete attribute map maps discrete values to graphical properties. A range attribute map maps numeric values or ranges of numeric values to graphical properties.
Defining a Discrete Attribute Map

How to Define a Discrete Attribute Map

You define a discrete attribute map in the following way:

- Specify the attribute mapping information in a SAS data set or in a DISCRETEATTRMAP block in the template code. See “Specifying the Attribute Mapping Information in a DISCRETEATTRMAP Block” on page 1414 and “Specifying the Attribute Mapping Information in a SAS Data Set” on page 1415.

Note: Defining a discrete attribute map in a SAS data set is valid in SAS 9.4M1 and later releases.

- Reference the attribute map in plot statements by using one of the following methods:
  - Include a DISCRETEATTRVAR statement in the template code to create an attribute map variable. The variable associates the attribute map with a classification column in the plot data. Then use the attribute map variable to reference the discrete attribute map in plot statements. This method applies to an attribute map that is defined in a SAS data set or in a DISCRETEATTRMAP block in the template code.
  - Specify the classification column, which is in the plot data, in the plot statements where needed. When the graph is rendered, the SGRENDER DATTRVAR statement is used to associate the attribute map that is defined in a SAS data set with the classification column in the plot data. This method applies only to an attribute map that is defined in a SAS data set.

Note: The DATTRVAR statement is valid in SAS 9.4M1 and later releases. For information about the DATTRVAR statement, see SAS ODS Graphics: Procedures Guide.

Specifying the Attribute Mapping Information in a DISCRETEATTRMAP Block

Use a DISCRETEATTRMAP block to specify the attribute mapping information in the template code. Place the block in the global definition area of your template between the BEGINGRAPH statement and the first layout statement. The block contains one or more VALUE statements that specify the attribute mapping information. Each VALUE statement specifies a single discrete value and one or more graphical properties that are assigned to that value. The NAME= option in the DISCRETEATTRMAP statement specifies the name of the attribute map. The
DISCRETEATTRVAR statement creates the reference variable for the attribute map. The following options are in the DISCRETEATTRVAR statement:

- The ATTRVAR= option specifies a unique name for the attribute-map-to-data-set-column association.
- The ATTRMAP= option specifies the value of the NAME= option that is specified in the DISCRETEATTRMAP statement.
- The VAR= option specifies the name of the numeric or character column in the plot data set, an expression, or the name of a dynamic variable.

Use the attribute map variable to reference the attribute map in plot statements.

**Note:** Do not use the attribute variable in an expression. Doing so might produce unexpected results.

**Note:** The values and graphical attributes that are defined in a discrete attribute map cannot be displayed by a CONTINUOUSLEGEND statement.

For more information, see “Defining a Discrete Attribute Map in a DISCRETEATTRMAP Block” on page 1430.

---

**Specifying the Attribute Mapping Information in a SAS Data Set**

Starting with **SAS 9.4M1**, you can specify the attribute mapping information for a discrete attribute map in a SAS data set instead of in a DISCRETEATTRMAP block in the template. You must store the attribute mapping information in a SAS data set. Attribute mapping information stored in a Cloud Analytic Services (CAS) in-memory table is not supported. Each observation in the data set maps a discrete value to one or more specific graphical properties. The following table lists the columns that you must include in your data set for each observation.

**Table 17.1  Columns That Are Required in Each Observation**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Character</td>
<td>Specifies the name of the attribute map to which this observation is associated. The ID value must be a SAS name token and not a literal. A data set can define multiple attribute maps. All observations that are associated with a specific attribute map must have the same ID value. When you are creating an attribute map variable for this attribute map, specify the name that you specified in the ID column in the DISCRETEATTRVAR statement ATTRMAP= argument. The ID column can also be specified in an SGRENDER DATTRVAR statement when the graph is rendered. The ID value is case sensitive.</td>
</tr>
</tbody>
</table>
The following table lists the columns that you can use to specify the graphical properties for each classification value. Properties that you do not specify default to the properties that are normally used when an attribute map is not specified.

**Table 17.2 Optional Columns That Specify Graphical Properties**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Character</td>
<td>Specifies the discrete classification value that is to be mapped to the graphical properties that are specified in this observation. The value is case sensitive.</td>
</tr>
<tr>
<td>FillColor</td>
<td>Character</td>
<td>Specifies the color for the filled areas for this classification value. When FillStyleElement is also specified, this column overrides the specified style element’s color attribute.</td>
</tr>
<tr>
<td>FillStyleElement</td>
<td>Character</td>
<td>Specifies the name of a style element, such as GraphData3, that is to provide the fill attributes.</td>
</tr>
<tr>
<td>FillTransparency</td>
<td>Numeric</td>
<td>Specifies the transparency of the fill color for this classification value. Values are in the range 0 (opaque) to 1 (completely transparent).</td>
</tr>
<tr>
<td>LineColor</td>
<td>Character</td>
<td>Specifies the line color for this classification value. When LineStyleElement is also specified, this column overrides the specified style element’s contrast color attribute.</td>
</tr>
<tr>
<td>LinePattern</td>
<td>Character</td>
<td>Specifies the line pattern for this classification value. You can specify the pattern name or number. See “Available Line Patterns” on page 1526.</td>
</tr>
<tr>
<td>LineStyleElement</td>
<td>Character</td>
<td>Specifies the name of a style element, such as GraphData3, that is to provide the line attributes.</td>
</tr>
<tr>
<td>LineThickness</td>
<td>Numeric</td>
<td>Specifies the line thickness, in pixels, for this classification value. The values must be integers.</td>
</tr>
<tr>
<td>MarkerColor</td>
<td>Character</td>
<td>Specifies the color of the marker symbol for this classification value. When MarkerStyleElement is also specified, this column overrides the specified style element’s contrast color attribute.</td>
</tr>
<tr>
<td>MarkerSize</td>
<td>Numeric</td>
<td>Specifies the size of the marker, in pixels, for this classification value. The values must be integers.</td>
</tr>
<tr>
<td>Column Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MarkerStyleElement</td>
<td>Character</td>
<td>Specifies the name of a style element, such as GraphData3, that is to provide the marker attributes.</td>
</tr>
<tr>
<td>MarkerSymbol</td>
<td>Character</td>
<td>Specifies the marker symbol to use for this classification value. See SYMBOL= in “Marker Options” on page 1523. When MarkerStyleElement is also specified, this column overrides the specified style element’s marker symbol attribute.</td>
</tr>
<tr>
<td>MarkerTransparency</td>
<td>Numeric</td>
<td>Specifies the transparency of the marker for this classification value. Values are in the range 0 (opaque) to 1 (completely transparent).</td>
</tr>
<tr>
<td>NoCase</td>
<td>Character</td>
<td>Specifies whether the attribute-map value comparisons are case-insensitive. Valid values are TRUE and FALSE. When TRUE, value comparisons are case insensitive. The default is TRUE. Note: This column is valid starting with SAS 9.4M3.</td>
</tr>
<tr>
<td>TextColor</td>
<td>Character</td>
<td>Specifies the text color for this classification value.</td>
</tr>
<tr>
<td>TextFont</td>
<td>Character</td>
<td>Specifies the text font by name for this classification value.</td>
</tr>
<tr>
<td>TextSize</td>
<td>Numeric</td>
<td>Specifies the text font size in points for this classification value. The values must be integers.</td>
</tr>
<tr>
<td>TextStyle</td>
<td>Character</td>
<td>Specifies the text style for this classification value. Valid values are NORMAL and ITALIC.</td>
</tr>
<tr>
<td>TextStyleElement</td>
<td>Character</td>
<td>Specifies the name of a style element, such as GraphLabelText, that is to provide the text attributes.</td>
</tr>
<tr>
<td>TextWeight</td>
<td>Character</td>
<td>Specifies the text weight for this classification value. Valid values are NORMAL and BOLD.</td>
</tr>
<tr>
<td>Show</td>
<td>Character</td>
<td>Specifies whether the legend displays only the attribute map values that appear in the data or always displays all of the values in the attribute map. Valid values are DATA and ATTRMAP. The default is DATA. Note: This column is valid starting with SAS 9.4M3.</td>
</tr>
</tbody>
</table>

The ID column provides the name of the attribute map, which is specified in the ATTRMAP= option in the DISCRETEATTRVAR statement or in a DATATTRVAR
Defining a Range Attribute Map

A range attribute map is defined as follows:

- The attribute mapping information is specified in a RANGEATTRMAP block in the template code. It must be placed in the global definition area of your template between the BEGINGRAPH statement and the first layout statement. The NAME= option in the RANGEATTRMAP statement specifies the name of the attribute map. The block contains one or more RANGE statements that specify the attribute mapping information. Each RANGE statement specifies a numeric value or numeric value range, and one or more graphical properties that are assigned to that value or range. An ENDRANGEATTRMAP statement is used to close the block.

  Note: Range attribute maps do not support attribute map information that is stored in a SAS data set.

- The RANGEATTRVAR statement is used to create an attribute map variable that associates the attribute map with a column in the data that provides the classification values. In the RANGEATTRVAR statement:
  - The ATTRVAR= option specifies a unique name for range attribute map to data set column association.
  - The ATTRMAP= option specifies the value of the NAME= option that is specified in the RANGEATTRMAP statement.
  - The VAR= option specifies the name of the numeric column in the plot data set with which the range attribute map is to be associated.

  Note: A RANGEATTRMAP can be used with a numeric column only.

- The attribute map variable is used to reference the discrete attribute map in plot statements.

  Note: The values and graphical attributes defined in a range attribute map cannot be displayed by a DISCRETELEGEND statement.

For information about how to reference a range attribute map, see “Referencing a Discrete Attribute Map in Plot Statements” on page 1419.

The RANGE statements in the RANGEATTRMAP block can associate a range of values or a single value with a single color or a color ramp. The syntax of the RANGE statement is as follows:

RANGE low-value < < > – < < > high-value / options
The optional exclusion operator (<) can be placed after the low value or before the high value to exclude that value from the range endpoint. The low value and high value can be an unformatted numeric value or a range keyword. For the low value, keyword MIN, NEGMAX, or NEGMAXABS can be used instead of numeric value. For the high value, keyword MAX or MAXABS can be used. For information about the range keywords, see SAS Graph Template Language: Reference.

**Note:** If two ranges share a common endpoint, such as 10–20 and 20–30, and no exclusion operator ( < ) is used, then the common endpoint belongs to the lower range, which is 10–20 in this case.

For more information about how to create a range attribute map, see “Creating and Using a Range Attribute Map” on page 1444.

---

Referencing an Attribute Map in Your Plot Statements

Referencing a Discrete Attribute Map in Plot Statements

If you use a DISCRETEATTRVAR statement to create a variable for an attribute map, then reference the attribute map in a plot statement by specifying the name that is assigned in the DISCRETEATTRVAR statement's ATTRVAR= argument. Each plot statement's documentation in this reference indicates which options support a reference to a discrete attribute map variable as the specified value.

**Note:** A reference to a discrete attribute variable must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

When you use a DATTRVAR statement in the SGRENDER statement, the attribute map is referenced implicitly when a column that is assigned to the attribute map ID in the DATTRVAR statement is referenced in a plot statement. The following table lists the statement options that support a reference to a discrete attribute map.

**Table 17.3  Statement Options That Support a Discrete Attribute Map**

<table>
<thead>
<tr>
<th>Statement</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXISTABLE</td>
<td>COLORGROUP= on page 208</td>
</tr>
<tr>
<td></td>
<td>TEXTGROUP= on page 216</td>
</tr>
<tr>
<td>BANDPLOT</td>
<td>GROUP= on page 232</td>
</tr>
<tr>
<td>BARCHART</td>
<td>GROUP= on page 258</td>
</tr>
<tr>
<td>BARCHARP Parm</td>
<td>GROUP= on page 301</td>
</tr>
<tr>
<td>Statement</td>
<td>Options</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>BOXPLOT</td>
<td>GROUP= on page 356</td>
</tr>
<tr>
<td>BOXPLOTPARM</td>
<td>GROUP= on page 394</td>
</tr>
<tr>
<td>BUBBLEPLOT</td>
<td>GROUP= on page 423</td>
</tr>
<tr>
<td>ELLIPSEPARM</td>
<td>GROUP= on page 486</td>
</tr>
<tr>
<td>FRINGEPLT</td>
<td>GROUP= on page 494</td>
</tr>
<tr>
<td>HEATMAPPARM</td>
<td>COLORGROUP= on page 514</td>
</tr>
<tr>
<td>HIGHLOWPLOT</td>
<td>GROUP= on page 536</td>
</tr>
<tr>
<td>LINECHART</td>
<td>GROUP= on page 536</td>
</tr>
<tr>
<td>LINEPARM</td>
<td>GROUP= on page 612</td>
</tr>
<tr>
<td>LOESSPLOT</td>
<td>GROUP= on page 623</td>
</tr>
<tr>
<td>MOSAICPLOTPARM</td>
<td>COLORGROUP= on page 642</td>
</tr>
<tr>
<td>NEEDLEPLOT</td>
<td>GROUP= on page 663</td>
</tr>
<tr>
<td>PBSPLINEPLOT</td>
<td>GROUP= on page 677</td>
</tr>
<tr>
<td>PIECHART</td>
<td>CATEGORY= on page 686</td>
</tr>
<tr>
<td>POLYGONPLOT</td>
<td>GROUP= on page 708</td>
</tr>
<tr>
<td>REGRESSIONPLOT</td>
<td>GROUP= on page 748</td>
</tr>
<tr>
<td>SCATTERPLOT</td>
<td>GROUP= on page 771</td>
</tr>
<tr>
<td>SCATTERPLOTMATRIX</td>
<td>GROUP= on page 808</td>
</tr>
<tr>
<td>SERIESPLOT</td>
<td>GROUP= on page 842</td>
</tr>
<tr>
<td>STEPPLOT</td>
<td>GROUP= on page 883</td>
</tr>
<tr>
<td>VECTORPLOT</td>
<td>GROUP= on page 943</td>
</tr>
<tr>
<td>WATERFALLCHART</td>
<td>COLORGROUP= on page 957</td>
</tr>
</tbody>
</table>

In a DISCRETELEGEND statement, reference the plot statement that uses the attribute map. The plot statement must have a NAME= option that assigns a name to the plot, because the DISCRETELEGEND statement references that name.
Because the attribute map is referenced in the plot statement, the legend uses the attribute map to represent the group values that exist in the data.

**TIP** If the discrete attribute map is defined in a DISCRETEATTRMAP block, then you can use the DISCRETELEGENDENTRYPOLICY=ATTRMAP option in the DISCRETEATTRMAP statement to display all of the items that are defined in the attribute map regardless of whether the values appear in the data. See DISCRETELEGENDENTRYPOLICY= on page 1426.

For an example of how to reference an attribute map variable that is created with the DISCRETEATTRVAR statement, see “Example: DISCRETEATTRVAR Statement with an Attribute Map Data Set” on page 1436.

The following code shows you how to reference an attribute map.

```plaintext
data attrds;
  input ID $1-7 VALUE $8-14 FILLCOLOR $15-26 LINECOLOR $27-38;
datalines;
  origin Asia   DodgerBlue  Black
  origin Europe Chartreuse  Black
  origin USA    SlateGray   Black
;run;

proc sort data=sashelp.cars out=cars;
  by origin;
run;

proc summary data=cars;
  class type;
  ways 1;
  by origin;
  output out=counts(drop=_TYPE_ rename=(_FREQ_=Count));
run;

/* Define the template for this graph */
proc template;
  define statgraph barchart;
    begingraph / attrpriority=none;
      entrytitle "Total Models Produced By Origin and Type";
      layout overlay /
        xaxisopts=(label="Vehicle Type")
        yaxisopts=(label="Total Models Produced");
      barchart category=type response=count / name="bar"
        display=all barlabel=true
        group=origin groupdisplay=cluster;
      discretelegend "bar";
    endlayout;
    endgraph;
  end;run;

proc sgrender data=counts datattrmap=attrds template=barchart;
  datattrvar origin="origin"
  where origin ne "USA";
run;
```
SAS data set Attrds is created to define attribute map ORIGIN. For each value of origin, attribute map ORIGIN specifies a fill color and a line color. The SUMMARY procedure is used to create data set Counts, which contains the counts of vehicle models by type and origin from table Sashelp.Cars. The TEMPLATE procedure defines a bar chart of columns Type and Count grouped by column Origin in table Counts. The SGRENDER procedure renders the graph.

In the SGRENDER procedure step, the DATTRMAP= option specifies the Attrds attribute map data set. The DATTRVAR statement associates discrete attribute map ORIGIN with classification column Origin in table Counts. A WHERE statement eliminates cars that are manufactured in the USA. The WHERE statement applies to the data in data set Counts only. It does not apply to the data in attribute map data set Attrds.

**TIP** To subset data in your attribute map data set using a WHERE clause, use a WHERE statement in a DATA step to create a new attribute map data set or use the data set WHERE option in the SGRENDER procedure DATTRMAP= option. Here is an example of the latter:

```plaintext
dattrmap=attrds(where=(value ne "USA"))
```

Here is the output for this example.

![Total Models Produced By Origin and Type](image)

Referencing a Range Attribute Map in Plot Statements

In a plot statement, you reference a range attribute map by specifying the name that is assigned in the RANGEATTRVAR statement's ATTRVAR= argument. Each plot statement's documentation in this reference indicates which options support a reference to a range attribute map variable as the specified value.
Note: A reference to a range attribute variable must be a direct reference to the attribute variable. It cannot be set by a dynamic variable.

The following table lists the statement options that support a reference to a range attribute map variable.

Table 17.4 Statement Options That Support a Range Attribute Map Variable Reference

<table>
<thead>
<tr>
<th>Statement</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUBBLEPLOT</td>
<td>COLORRESPONSE=</td>
</tr>
<tr>
<td>HEATMAP</td>
<td>COLORRESPONSE=</td>
</tr>
<tr>
<td>MOSAICPLOTPARM</td>
<td>COLORRESPONSE=</td>
</tr>
<tr>
<td>POLYGONPLOT</td>
<td>COLORRESPONSE=</td>
</tr>
<tr>
<td>SCATTERPLOT</td>
<td>COLORRESPONSE= or MARKERCOLORGRADIENT=</td>
</tr>
<tr>
<td>SCATTERPLOTMATRIX</td>
<td>COLORRESPONSE= or MARKERCOLORGRADIENT=</td>
</tr>
<tr>
<td>SURFACEPLOTPARM</td>
<td>COLORRESPONSE= or SURFACECOLORGRADIENT=</td>
</tr>
<tr>
<td>WATERFALLCHART</td>
<td>COLORRESPONSE=</td>
</tr>
</tbody>
</table>

In a CONTINUOUSLEGEND statement, reference the plot statement that uses the attribute map. The plot statement must have a NAME= option that assigns a name to the plot, because the CONTINUOUSLEGEND statement references that name. Because the attribute map is referenced in the plot statement, the legend uses that attribute map to represent the numeric values that are present in the data.

For an example, see “Example 1: Using a Range Attribute Map to Control Marker Colors” on page 1445.
Discrete Attribute Map Statements

Dictionary

DISCRETEATTRMAP Statement

Defines a set of graphical properties that can be associated with user-defined sets of values.

Requirements:
You must place the DISCRETEATTRMAP block directly inside the BEGINGRAPH block between the BEGINGRAPH statement and the first LAYOUT statement. Do not embed it in any other GTL statement block.

The DISCRETEATTRMAP block must contain at least one VALUE statement.

The DISCRETEATTRVAR statement must be used to associate the discrete attribute map with a data column.

Notes:
The graphical properties for a discrete attribute map can also be defined in a SAS data set. See “Defining a Discrete Attribute Map” on page 1414.

Prior to the third release for SAS 9.4, when a discrete attribute map is used for group values in a plot that contributes to a discrete legend and attributes are overridden in the plot statement, the attributes of some plot features and their corresponding legend items might not match. Starting with SAS 9.4M3, the attributes of the legend items always match the attributes of the corresponding plot features.

See:
“DISCRETEATTRVAR Statement” on page 1435

Syntax

```
DISCRETEATTRMAP NAME="string" </options>;
    VALUE value-spec </options>;
    <... more VALUE statements ...>
ENDDISCRETEATTRMAP;
```

1425
Required Argument

NAME="string"

assigns a name to the attribute definition. The name can be referenced in a DISCRETEATTRVAR statement, which is used to associate the attribute map with an input data column. The name can also be referenced in a DISCRETELEGEND statement to map the specified graphical properties directly to a discrete legend.

Restrictions

The string is case sensitive, must be enclosed in quotation marks, and must define a unique name within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

Optional Arguments

DISCRETELEGENDENTRYPOLICY=DATA | ATTRMAP

specifies whether the items that are contributed to a discrete legend by the plot associated with this attribute map are only items that appear in the data or only items that are defined in the attribute map.

Note: This option is valid starting in SAS 9.4M1.

DATA

the associated plot contributes to a discrete legend only items that appear in the data.

ATTRMAP

the associated plot contributes to a discrete legend only items that are defined in the discrete attribute map.

Interaction

If this option is set to ATTRMAP, then data skins, overrides from the DATATRANSPARENCY= option, and overrides from the TRANSPARENCY= suboption in the FILLATRrs= and MARKERATRrs= options are displayed in the discrete legend. Overrides from other options such as the COLOR= suboption in the FILLATRrs= option are not displayed.

Default DATA

IGNORECASE=TRUE | FALSE

specifies whether case is ignored when comparing the values that are specified in the attribute map with values from an input data column.

Default FALSE. Value comparisons are case-sensitive.

Tip

The effect of this option can be achieved by applying a function like UPCASE to the data column and using only uppercase strings in each VALUE statement.

See “boolean” on page 1479 for other Boolean values that you can use.

TRIMLEADING=TRUE | FALSE

specifies whether leading blanks are trimmed (removed) from both the attribute map values and the input data values before those values are compared. Trailing blanks are always trimmed.
Default **TRUE.** Leading blanks are trimmed.

See “**boolean**” on page 1479 for other Boolean values that you can use.

**VALUE Statement Required Argument**

**value-spec**

specifies one or more formatted strings or the keyword **OTHER.** Strings are always quoted. Multiple strings must be separated by blanks, and each of the strings must be enclosed in its own set of quotation marks. The formatted strings must be equal to the formatted values of the classification column that is used with the **DISCRETEATTRVAR** statement.

**OTHER**

creates a category for all other column values that are not explicitly assigned with **VALUE** statements. This keyword is not quoted. The default attributes for these values are derived from the GraphOther style element.

**Note** If **OTHER** is not specified, then data values that are not explicitly assigned with **VALUE** statements are mapped to attributes as if a discrete attribute map is not in effect.

The following examples elaborate on the **value-spec** strings:

"Hybrid"

By default, all string comparisons are case-sensitive. By default, the string **Hybrid** does not match the string **HYBRID.**

"HYBRID"

If **IGNORECASE=TRUE** in the **DISCRETEATTRMAP** statement, then you can specify an upper-, lower-, or mixed-cased string for the **value-spec** string. When **IGNORECASE=TRUE**, the string **Hybrid** matches the string **HYBRID.**

"15JAN2011"

If a numeric column is being mapped with a **VALUE** statement, then you must specify the formatted value of the column. This example shows how to specify the value of a numeric SAS date column that has a DATE9. format associated with it.

"."

If a numeric column has a missing value, then you should use the formatted value for missing, which is "." by default. If the **MISSING=** system option is used to change the default string, then you should match that value. For example, if **OPTIONS MISSING="M"** is specified in the SAS program, then you should use "M" in the **VALUE** statement to represent missing values.

" "

If a character column has a missing value, then you should use the formatted value for missing, which is " " by default.

"Truck" "SUV"

Multiple strings can be specified to indicate that each of the specified values matches to the same graphical properties. It does not mean that a single new category is formed. The list of strings is separated by blanks, and each string is enclosed in its own set of quotation marks.

**Restrictions** A dynamic variable reference or run-macro variable reference cannot be used in **value-spec.**
If a user-defined format is associated with the classification column, then you should specify the same formatted strings that appear in the format definition.

**Note**
When the specified value strings are compared with the data values, leading blanks are honored and trailing blanks are ignored.

**Tip**
In SAS 9.4M1 and earlier releases, if you create and use a format to display a special value for missing character values, in some cases, " " is returned instead of the formatted value. If your attribute map assigns attributes to the formatted missing value in that case, the attributes are not assigned to the missing values. To correct this problem, specify both " " and your formatted missing value in the VALUE statement for your missing-value attributes. This issue is resolved in SAS 9.4M2.

### VALUE Statement Optional Arguments

The following options can be used in the VALUE statement.

**FILLATTRS=**

- **style-element | style-element (fill-options) | (fill-options)**
  
  the fill attributes to be used when an attribute map is applied to filled areas in a graph.

  **Defaults**
  
  When DISCRETELEGENDENTRYPOLICY=DATA, unspecified attributes receive the attributes that they would have if the attribute map were not defined.

  When DISCRETELEGENDENTRYPOLICY=ATTRMAP, unspecified attributes derive attributes from the GraphDataDefault style element.

  **Restriction**
  
  In SAS 9.4 and earlier releases, the TRANSPARENCY= fill option is ignored. Starting with SAS 9.4M2, the TRANSPARENCY= fill option is supported.

  **Requirement**
  
  When you specify TRANSPARENCY, you must also specify COLOR. Otherwise, TRANSPARENCY is ignored.

  **See**
  
  "General Syntax for Attribute Options" on page 1519 for the syntax on using a style-element.

  "Fill Color Options" on page 1520 for available fill-options.

**LINEATTRS=**

- **style-element | style-element (line-options) | (line-options)**
  
  specifies the line attributes to be used when an attribute map is applied to lines in a graph.

  **Defaults**
  
  When DISCRETELEGENDENTRYPOLICY=DATA, unspecified attributes receive the attributes that they would have if the attribute map were not defined.

  When DISCRETELEGENDENTRYPOLICY=ATTRMAP, unspecified attributes derive attributes from the GraphDataDefault style element.
Restrictions
In SAS 9.4 and earlier releases, the THICKNESS= line option is ignored. Starting with SAS 9.4M1, the THICKNESS= line option is supported.

Line option PATTERN= is ignored for the bar outlines in histograms.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“LINE OPTIONS” on page 1522 for available line-options.

MARKERATTRS=\text{style-element} | \text{style-element (marker-options)} | (marker-options)
specifies the marker attributes to be used when an attribute map is applied to marker symbols in a graph.

Defaults
When DISCRETELEGENDENTRYPOLICY=DATA, unspecified attributes receive the attributes that they would have if the attribute map were not defined.

When DISCRETELEGENDENTRYPOLICY=ATTRMAP, unspecified attributes derive attributes from the GraphDataDefault style element.

Restrictions
The WEIGHT= marker option is ignored.

In SAS 9.4 and earlier releases, the SIZE=, TRANSPARENCY=, and WEIGHT= marker options are ignored. Starting with SAS 9.4M1, the SIZE=, TRANSPARENCY=, and WEIGHT= marker options are supported.

For filled, outlined markers, COLOR and TRANSPARENCY are ignored.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Marker Options” on page 1523 for available marker-options.

TEXTATTRS=\text{style-element} | \text{style-element (text-options)} | (text-options)
specifies the text attributes to use when an attribute map is applied to text in a graph.

Default
The GraphDataText style element.

Restriction
This option is honored only by the TEXTGROUP= option in the AXISTABLE statement and the GROUP= option in the TEXTPLOT statement.

See
“General Syntax for Attribute Options” on page 1519 for the syntax on using a style-element.

“Text Options” on page 1525 for available text options.
Details

About Discrete Attribute Maps

Attribute maps can be useful for ensuring that a particular value (a company name, for example) is always represented by the same visual characteristics in your graphs, regardless of the value’s order in the input data. When specified directly in a discrete legend, an attribute map can be used to display legend entries for group values that are not in the data. (See “Displaying Legend Entries for Group Values That Are Not in the Data” on page 1431.)

If you do not want to manage the graphical properties that are associated with each unique discrete value, then you can simply specify an input column in the grouping option. In that case, each discrete value is represented by different graphical properties in the graph. The default properties are derived from options that are set in the plot statement or from the GraphData1–GraphDataN style elements that are defined in the ODS style that is in effect. However, to ensure that specific graphical properties are used to represent a discrete value in the graph, regardless of that value’s order in the data, you can use an attribute map to create that association. Any values in the data that are not accounted for in the attribute map are assigned the graphical properties that they would receive if the attribute map is not defined.

Defining a Discrete Attribute Map in a DISCRETEATTRMAP Block

A DISCRETEATTRMAP block creates an attribute map that matches graphical properties to discrete values. The attribute map can be associated with a data input column that is used as a classification variable in a graph. It can also be specified directly in a discrete legend.

To define a discrete attribute map in a DISCRETEATTRMAP block, do the following:

- Use the DISCRETEATTRMAP statement to start the attribute definition and assign a name to it. The DISCRETEATTRMAP statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in another DISCRETEATTRMAP statement or in a RANGEATTRMAP statement. The DISCRETEATTRMAP statement determines whether the data mapping is case-sensitive and whether leading blanks are trimmed from the data values during the mapping.

- Nest within the DISCRETEATTRMAP block at least one VALUE statement that specifies graphical properties to associate with a classification value. See “About the VALUE Statement” on page 1430. Use one VALUE statement for each classification value that you want to map. Values that are referenced in the attribute map can be character or numeric. Discrete values that are not accounted for in the attribute map are assigned the graphical properties that they would receive if the attribute map is not defined.

- Use the ENDDISCRETEATTRMAP statement to close the block.

For information about how to use the attribute map, see “Using a Discrete Attribute Map” on page 1431.

About the VALUE Statement

A VALUE statement within the DISCRETEATTRMAP block associates graphical properties with a discrete value in the attribute map. To associate graphical properties with multiple values, specify multiple VALUE statements in the attribute map using the following general syntax for each of the statements:

```
VALUE value-spec <options>
```

For a specific example, see the “Example Program” on page 1434.

If the discrete attribute map is referenced by a plot statement in the template, then the graphical properties that are defined in the VALUE statements are used in the plot. If a discrete legend is generated for the plot, then the graphical properties are represented in that legend.

If a discrete attribute map is referenced directly in a DISCRETELEGEND statement, then the graphical properties that are defined in the VALUE statement are mapped directly to the legend and are independent of the values in the data. For more information and an example, see the “Defining a Discrete Attribute Map in a DISCRETEATTRMAP Block” on page 1430.

If two or more VALUE statements define attributes to associate with the same classification values, then the last VALUE statement's settings are used.

By default when comparing a column's value to a string that is specified for the VALUE statement's value-spec,

- the column value is formatted to a string, using the format that is defined for the column or the default format if no format is defined for the column
- leading spaces are trimmed from the string that is specified in the VALUE statement
- a case-sensitive comparison is performed between the column string and the VALUE string.

To change the default behavior for the comparison, you can use the DISCRETEATTRMAP statement's TRIMLEADING= and IGNORECASE= options.

Using a Discrete Attribute Map

To use a discrete attribute map that is defined in a DISCRETEATTRMAP block, do the following:

- Use the DISCRETEATTRVAR statement to create a named association between the defined attribute map and the input column that contains the classification values. The ATTRMAP= argument identifies the attribute map and the VAR= argument identifies the input column. Use the ATTRVAR= argument to assign a name that can be used to reference the named association in plot statements within the template.
- Reference the attribute map variable in the plot statements where needed. See “Referencing a Discrete Attribute Map in Plot Statements” on page 1419.

For an example, see “Example Program” on page 1434.

Displaying Legend Entries for Group Values That Are Not in the Data

To display legend entries for the values in a grouped plot, you typically use the plot statement's NAME= option to assign a name to the plot, and then reference that name in the DISCRETELEGEND statement. In this usage case, the legend displays entries for the group values that exist in the data.

To represent all of the group values in the legend, regardless of whether they exist in the data, you can specify DISCRETELEGENDENTRYPOLICY=ATTRMAP in the DISCRETEATTRMAP statement for the attribute map. When the DISCRETELEGENDENTRYPOLICY=ATTRMAP option is in effect, the associated plot contributes all of the items in the attribute map to the discrete legend regardless of whether they exist in the data.
Displaying all of the items in a discrete attribute map in the legend can be useful for flagging data in the graph. For example, assume you have weight and height values for all students in an analysis group and you want to create a scatter plot of the data. However, some of the observations are incomplete and do not record the student’s sex. You want to include the incomplete observations in your plot, but you want to visually distinguish them from the others. In that case, you can do the following:

- Represent the unknown values in your data by entering the value U for sex.
- Define the discrete attribute map for the plot in a DISCRETEATTRMAP block. Include the DISCRETELEGENDENTRYPOLICY=ATTRMAP option in the DISCRETEATTRMAP statement. Include VALUE statements that specify the properties for the M, F, and U values.
- Use the DISCRETEATTRVAR statement to create a discrete attribute map variable that associates the attribute map with column Sex in the input data.
- In the SCATTERPLOT statement, specify the discrete attribute map variable name in the GROUP= option.
- In the DISCRETELEGEND statement, reference the scatter plot.

As a result, the legend displays the attribute-map definitions, and observations with the value U in column Sex are displayed as incomplete observations.

Here is an example of a template that uses a discrete attribute map to uniquely display observations with the value U in column Sex.

```proc template;
  define statgraph discreteattrmapdatapresent;
  begingraph;
    entrytitle "Height and Weight by Sex";
    /* Define the attribute map and assign the name "symbols" */
    discreteattrmap name="symbols" / trimleading=true ignorecase=true
discretelegendentrypolicy=attrmap;
    value "M" / markerattrs=(color=blue symbol=diamondfilled);
    value "F" / markerattrs=(color=green symbol=circlefilled);
    value "U" / markerattrs=(color=red symbol=starfilled);
    enddiscreteattrmap;

    /* Create attribute map variable GROUPMARKERS to associate attribute
      map SYMBOLS with column Sex */
    discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols";

    /* Use the attribute map by referencing GROUPMARKERS in a plot
      statement */
    layout overlay;
      scatterplot x=height y=weight / name="scatter"
group=groupmarkers;
      discretelegend "scatter";
    endlayout;
  endgraph;
end;
```

To test the template, you can generate test data set Testclass from Sashelp.Class by changing the value of column Sex to U for John and Carol as shown in the following code.

```data testclass;
  set sashelp.class;
```
if (name="John") then sex="U";
if (name="Carol") then sex="U";
run;

The following figure shows the output of this template when it is run with the test data set Testclass.

![Graph showing Height and Weight by Sex](image1)

The two red stars in the plot indicate the observations with the value U in column Sex. The next figure shows the output of this template when it is run with the complete data in data set Sashelp.Class.

![Graph showing Height and Weight by Sex](image2)

The absence of the red stars in the plot indicates that the value U is no longer present in the data. All of the observations are now complete. When DISCRETELEGENDENTRYPOLICY=ATTRMAP is specified for an attribute map, be aware that the legend entries that are contributed by an associated plot are defined entirely by the attribute map and are independent of the data.

**Example: DISCRETEATTRMAP and DISCRETEATTRVAR Statements**

The following graph was generated by the "Example Program" on page 1434. The example defines graphical properties to associate with classification values in an input column that is used in a scatter plot. The DISCRETEATTRMAP statement starts the attribute map definition, assigns a name to it, and ensures that the data mapping is not case sensitive. The VALUE statements define the colors and marker symbols to associate with the values M and F. The DISCRETEATTRVAR statement associates the attribute map with the data column Sex and assigns the name GROUPMARKERS to the association. The SCATTERPLOT statement references the named association in its GROUP= option.
Example Program

```text
proc template;
  define statgraph scatterplot;
    begingraph;
      entrytitle "Height and Weight by Sex";

      /* define the attribute map and assign the name "symbols" */
      discreteattrmap name="symbols" / ignorecase=true;
        value "m" / markerattrs=(color=blue symbol=diamondfilled);
        value "f" / markerattrs=(color=red symbol=circlefilled);
      enddiscreteattrmap;

      /* associate the attribute map with input data column Sex and assign * the name GROUPMARKERS to the named association */
      discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols";

      /* reference GROUPMARKERS in a plot statement */
      layout overlay;
        scatterplot x=height y=weight / name="scatter"
          group=groupmarkers;
        discretelegend "scatter";
      endlayout;
    endgraph;
end;

proc sgrender data=sashelp.class template=scatterplot;
run;
```
DISCRETEATTRVAR Statement

Creates a named association between a user-defined discrete attribute map and an input data column.

**Requirements:**
- You must place the DISCRETEATTRVAR statement directly inside the BEGINGRAPH block between the BEADINGRAPH statement and the first LAYOUT statement. Do not embed it in any other GTL statement block.
- A discrete attribute map must be created using the DISCRETEATTRMAP statement.

**See:**
- “Example: DISCRETEATTRMAP and DISCRETEATTRVAR Statements” for an example.

**Syntax**

```plaintext
DISCRETEATTRVAR ATTRVAR=attrvar-name
VAR=data-column | expression | dynamic
ATTRMAP="attrmap-name";
```

**Required Arguments**

- `ATTRVAR=attrvar-name`
  - specifies a SAS name for this association between the attribute map and the input column. This name must be unique within the template and can be referenced by other statements that can be associated with the attribute map. The attribute map variable name should not be used in an expression. If it is, then the results are unpredictable.

**Restrictions**

The name that is assigned in this argument is used to associate an attribute map with the discrete values in an input data column. Thus, it is not the name to reference when you want to display legend entries that are independent of the data. For that special use, a DISCRETELEGEND statement can reference the attribute map directly by the name that is assigned in the DISCRETEATTRMAP statement. For more information, see the DISCRETEATTRMAP statement’s “Defining a Discrete Attribute Map in a DISCRETEATTRMAP Block” on page 1430.

Using an attribute variable in an expression is not supported.

**Note**

- The assigned SAS name can be the same as the name of the data input column, but it is not recommended. If an assigned `attrvar-name` matches the name of an input data column, then the `attrvar-name` takes precedence.

- `VAR=data-column | expression | dynamic`
  - specifies an input data column to be associated with an attribute map at run time. If an `expression` is used, a new column of transformed values is created and then matched with the attribute map.

**Interaction**

- If the column is not found or the column is of the wrong type for the attribute map, then the DISCRETEATTRVAR statement is ignored.
Tip
The input data column can be character or numeric, but the values must match the type of the values that are specified in the attribute map. For numeric columns, all values are treated as discrete values.

**ATTRMAP=**"**attrmap-name**"

specifies the name of an existing discrete attribute map.

Restriction
The **attrmap-name** is case sensitive, must be enclosed in quotation marks, and must be the name that was assigned to the attribute map in the DISCRETEATTRMAP statement’s NAME= argument.

Details
The DISCRETEATTRVAR statement creates and names an association between graphical properties that are specified in a DISCRETEATTRMAP block and a classification column that is in the data. The name that is assigned to the association in the DISCRETEATTRVAR statement is the name that plot statements must reference to use the attribute map.

Defining and using a discrete attribute map requires you to coordinate settings on several statements. For more information, see the DISCRETEATTRMAP statement’s “Defining a Discrete Attribute Map in a DISCRETEATTRMAP Block” on page 1430.

The DISCRETEATTRVAR statement must be located within the BEINGRAPH block but outside of the outermost layout block. It cannot be nested in a DISCRETEATTRMAP statement.

Example: DISCRETEATTRVAR Statement with an Attribute Map Data Set

Starting with **SAS 9.4M1**, you can define an attribute map in a SAS data set. This example shows you how to use the DISCRETEATTRVAR statement with an attribute map that is defined in a SAS data set. It is the example in “Example: DISCRETEATTRMAP and DISCRETEATTRVAR Statements” on page 1433 modified to use a SAS data set instead of a DISCRETEATTRMAP block.

Here is the output for this example.
Example Program

Here is the SAS code for this example.

/* Create the attribute map data set */
data attrs;
   input ID $1-7 VALUE $9 MARKERSYMBOL $11-23 MARKERCOLOR $25-30;
datalines;
symbols M diamondfilled blue
symbols F circlefilled red
;run;

/* Define the template for this graph */
proc template;
define statgraph scatterplot;
begingraph;
   entrytitle "Height and Weight by Sex";
   discreteattrvar attrvar=groupmarkers var=sex attrmap="symbols";
endgraph;
end;

/* Reference GROUPMARKERS in a plot statement */
layout overlay;
scatterplot x=height y=weight / name="scatter"
   group=groupmarkers;
discretelegend "scatter";
endlayout;
endgraph;
end;
Details

The attribute map is defined in data set Attrds. The data set includes the required columns ID and VALUE, which specify a name for this attribute map and the value of the classification variable that is to be mapped. The MarkerSymbol and MarkerColor columns specify the graphical properties that are mapped to each value. The values specified in the Value column are case-sensitive. Unlike the DISCRETEATTRMAP statement, when you define your attribute map in a data set, there is no option that you can specify to ignore case. You must ensure that the case of the values in your attribute map matches the case in the actual data.

The DISCRETEATTRVAR statement associates the attribute map with the data column Sex and assigns the name GROUPMARKERS to the attribute map variable. The ATTRMAP= option specifies the name in the ID column of the attribute map data set. The value is case-sensitive. The attribute map variable name is specified in the GROUP= option in the SCATTERPLOT statement.

Finally, in the SGRENDER statement, the DATTRMAP= option specifies the name of the attribute map data set. When your attribute map is defined in a SAS data set, you must specify the name of the data set in the DATTRMAP= option in the SGRENDER statement that renders the graph.

You can also use the DATTRVAR= statement with the SGRENDER statement to associate the attribute map with the data column Sex. See "Referencing an Attribute Map in Your Plot Statements" on page 1419.
Range Attribute Map Statements

Dictionary

RANGEATTRMAP Statement

Creates an attribute map that matches colors to numeric values or numeric ranges so that the colors can be associated with the values of an input data column.

Restriction: A RANGEATTRMAP cannot be directly referenced in a legend statement.

Requirements: You must place the RANGEATTRMAP block directly inside the BEGINGRAPH block between the BEGINGRAPH statement and the first LAYOUT statement. Do not embed it in any other GTL statement block.
The RANGEATTRMAP block must contain at least one RANGE statement.
The RANGEATTRVAR statement must be used to associate the range attribute map with a data column.

Note: The RANGEATTRMAP statement defines a set of graphical properties for ranges of data values.

See: “RANGEATTRVAR Statement” on page 1448

Syntax

RANGEATTRMAP NAME="string";
    RANGE range-spec </options>;
    <… more-RANGE statements …>
ENDRANGEATTRMAP;

Required Argument

NAME="string"
    assigns a name to the attribute definition for reference in a RANGEATTRVAR statement.
Restrictions

The string is case sensitive, must be enclosed in quotation marks, and must define a name that is unique among RANGEATTRMAP names within the template.

This option does not support variables that are created by the DYNAMIC, MVAR, and NMVAR template statements.

RANGE Statement Required Argument

range-spec

specifies a range of numeric values or a keyword, such as OTHER or MISSING.

A range of numeric values is specified in the form low-value - high-value. Both the low value and the high value can be specified as an unformatted numeric value. A less-than symbol (<) can be placed after the low numeric value, before the high numeric value, or in both positions to exclude that value from the range endpoint (similar to the VALUE statement of PROC FORMAT). If you are excluding the first value in a range, then put the < after the low value. If you are excluding the last value in a range, then put the < before the high value. You can also exclude both the low and the high value.

For example, the following range does not include 0:

0 < - 100

Likewise, the following range does not include 100:

0 - < 100

If a value at the high end of one range also appears at the low end of another range and you do not use the < exclusion notation, then the value is assigned to the first range.

If two or more RANGE statements define colors to associate with the same numeric values or ranges, then the first RANGE statement's settings are used. If any RANGE statement's range overlaps another RANGE statement's range (for example, 10 - 20 and 15 - 25), then the entire attribute map is ignored and default coloring is used.

If two ranges share a common endpoint (for example, 10 - 20 and 20 - 30) and no exclusion operator is used, then the common endpoint belongs to the lower encountered range (10 - 20 in this case). The order of the specification does not matter.

To set a single numeric value, specify the same value for both the low value and the high value.

If the low value is not less than or equal to the high value, then the range specification is invalid, and the RANGE statement is ignored in the attribute map definition.

Note: If a range is not defined for keyword OTHER, then gaps within the attribute map ranges are assigned the default color that is defined by the GraphOther:ContrastColor style reference.

Rather than using a numeric value, you can specify one of the following keywords as the low value or the high value:

MIN indicates the minimum data value for column values.
MAX indicates the highest data value for the column values.
MAXABS indicates max(abs(MIN), abs(MAX))
NEGMAX indicates -MAX
NEGMAXABS indicates -max(abs(MIN), abs(MAX))

Rather than specifying a low-value-to-high-value range, you can use one of the following keywords for the range specification:

MISSING indicates a mapping for missing values. The visual attributes for this setting are obtained from the GraphMissing style element. If one RANGE statement specifies this value and another RANGE statement specifies keyword OTHER, then the OTHER range does not include missing values.

UNDER creates a range for all data values between the lowest mapped value and the lowest actual data value. The visual attributes for this setting are obtained from the GraphUnderflow style element. If one RANGE statement specifies this value and another RANGE statement specifies keyword OTHER, then the OTHER range does not include underflow values.

OVER creates a range for all data between the highest mapped value and the highest actual data value. The visual attributes for this setting are obtained from the GraphOverflow style element. If one RANGE statement specifies this value and another RANGE statement specifies keyword OTHER, then the OTHER range does not include overflow values.

OTHER creates a category for all other column values not explicitly assigned to a range. The OTHER values can be composed of several non-contiguous ranges. The visual attributes for this setting are obtained from the GraphOther style element.

Restriction A dynamic variable reference or run-time macro variable reference cannot be used in range-spec.

RANGE Statement Optional Arguments

RANGEALTCOLOR=style-reference | color |

GRADIENTSTEPPER(color1, color2, num-steps, step)

specifies a contrast color for lines and markers to represent the defined value range.

GRADIENTSTEPPER (color1, color2, num-steps, step)
a gradient stepper that partitions a color range into equal-sized intervals and returns the color that is in the specified step position. The start and end colors for the range are specified in parameters color1 and color2. The number of equal-sized intervals is specified in parameter num-steps, and the step position for the color to return is specified in parameter step.

Example:

rangeattrmap name="incomemap";
range min - 13000 / rangealtcolor=gradientstepper(red,green,4,1);
range 13000 < - 25000 / rangealtcolor=gradientstepper(red,green,4,2);
range 25000 < - 50000 / rangealtcolor=gradientstepper(red,green,4,3);
range 50000 < - max / rangealtcolor=gradientstepper(red,green,4,4);
endrangeattrmap;

Default  The GraphDataDefault:ContrastColor style reference.

Restriction A dynamic variable reference or run-time macro variable reference can be used to specify individual token values, but it cannot be used to replace the entire list of tokens.

Interaction  If this option is specified, then the RANGEALTCOLORMODEL= option is ignored.

Tips  Because lines and markers have less surface area, choose a contrasting color in order to improve visibility.

Use RANGECOLOR= to control fill color.

RANGECOLOR=style-reference | color | GRADIENTSTEPPER(color1,color2, num-steps, step )

specifies a color for area fill to represent the defined value range.

GRADIENTSTEPPER (color1,color2, num-steps, step )

a gradient stepper that partitions a color range into equal-sized intervals and returns the color that is in the specified step position. The start and end colors for the range are specified in parameters color1 and color2. The number of equal-sized intervals is specified in parameter num-steps, and the step position for the color to return is specified in parameter step. Example:

rangeattrmap name="incomemap";
range min - 13000 / rangecolor=gradientstepper(red,blue,4,1);
range 13000 < - 25000 / rangecolor=gradientstepper(red,blue,4,2);
range 25000 < - 50000 / rangecolor=gradientstepper(red,blue,4,3);
range 50000 < - max / rangecolor=gradientstepper(red,blue,4,4);
endrangeattrmap;

Default  The GraphDataDefault:ContrastColor style reference.

Restriction A dynamic variable reference or run-time macro variable reference can be used to specify individual token values, but it cannot be used to replace the entire list of tokens.

Interaction  If this option is specified, then the RANGEALTCOLORMODEL= option is ignored.

Tip  Use RANGEALTCOLOR= to control line and marker color.

RANGEALTCOLORMODEL=style-element | (list-of-colors)

specifies either a style element or a list of one or more specific contrast colors for lines and markers to represent the defined value range in this argument.

style-element

specifies the name of a style element. To display the range as a gradient ramp, choose a style element such as TwoColorRamp, TwoColorAltRamp, ThreeColorRamp, or ThreeColorAltRamp. The style element should contain the following style attributes:

STARTCOLOR  specifies a color for the smallest data value.
NEUTRALCOLOR  specifies a color for the midpoint of the data range. This attribute is not needed when defining a two-color ramp.
ENDCOLOR specifies a color for the highest data value.

To display the range or single value contrast color as a color that is defined in a style, use a style-reference (for example, GraphData1:color) to refer to a color attribute. The following style references correspond to the keywords that are available in this statement’s range-spec argument:

<table>
<thead>
<tr>
<th>range-spec Keyword</th>
<th>Corresponding Style Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISSING</td>
<td>GraphMissing:ContrastColor</td>
</tr>
<tr>
<td>OTHER</td>
<td>GraphOther:ContrastColor</td>
</tr>
<tr>
<td>UNDER</td>
<td>GraphUnderflow:ContrastColor</td>
</tr>
<tr>
<td>OVER</td>
<td>GraphOverflow:ContrastColor</td>
</tr>
</tbody>
</table>

(list-of-colors)

a space-separated list of two or more color keywords that is enclosed in parentheses.

Two colors create the endpoints of a ramp. The first color is assigned to the low value in the range specification, and the second color is assigned to the high value. Three or more colors partition the range specification into \( n-1 \) equal-sized intervals where each adjacent color pair defines a two-color ramp.

Interaction This option is ignored if the RANGEALTCOLOR= option is specified.

Tips Because lines and markers have less surface area, choose contrasting colors in order to improve visibility.

Use RANGECOLORMODEL= to control fill color.

RANGECOLORMODEL=style-element | (list-of-colors)

specifies either a style element or a list of one or more specific colors for area fills to represent the defined value range in this range-spec argument.

style-element

specifies the name of a style element. To display the range as a gradient ramp, choose a style element such as TwoColorRamp, TwoColorAltRamp, ThreeColorRamp, or ThreeColorAltRamp. The style element should contain the following style attributes:

<table>
<thead>
<tr>
<th>Style Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARTCOLOR</td>
<td>specifies a color for the smallest data value.</td>
</tr>
<tr>
<td>NEUTRALCOLOR</td>
<td>specifies a color for the midpoint of the data range. This attribute is not needed when defining a two-color ramp.</td>
</tr>
<tr>
<td>ENDCOLOR</td>
<td>specifies a color for the highest data value.</td>
</tr>
</tbody>
</table>

To display the range or single value as a color that is defined in a style, use a style-reference (for example, GraphData1:color) to refer to a color attribute. The following style references correspond to the keywords that are available in this statement’s range-spec argument:
**range-spec Keyword** | **Corresponding Style Reference**
--- | ---
MISSING | GraphMissing:ContrastColor
OTHER | GraphOther:ContrastColor
UNDER | GraphUnderflow:ContrastColor
OVER | GraphOverflow:ContrastColor

(list-of-colors)

A space-separated list of two or more color keywords that is enclosed in parentheses.

Two colors create the endpoints of a ramp. The first color is assigned to the low value in the range specification, and the second color is assigned to the high value. Three or more colors partition the range specification into \( n-1 \) equal-sized intervals, where each adjacent color pair defines a two-color ramp.

**Interaction**

This option is ignored if the RANGECOLOR option is specified.

**Tip**

Use RANGEALTCOLORMODEL= to control line and marker color.

---

**Details**

**Creating and Using a Range Attribute Map**

The RANGEATTRMAP statement creates an attribute map that matches colors to numeric values or value ranges. The attribute map can be associated with a data input column that uses color to represent response values in a graph. Attribute maps can be useful for controlling the application of gradient color in a graph. In addition, they enable you to map color to data values, independent of the actual data that is used in the graph. For example, for temperature data, you can set Blue for 0 and Red for 100, even if the values 0 and 100 are not in the data.

Defining and using the attribute map requires you to coordinate settings on several statements:

- Use the RANGEATTRMAP statement to start the attribute definition and assign a name to it.
- Nest within the RANGEATTRMAP block at least one RANGE statement that specifies a numeric value or numeric range and the color, contrast color, or both, to associate with that value or range. Use one RANGE statement for each value range that you want to map. You can use keywords like MIN and MAX in the range specification. For complete details about the range specifications, see the RANGE statement.
- Use the RANGEATTRVAR statement to create a named association between the defined attribute map and the input column that contains the numeric values. The ATTRMAP= argument identifies the attribute map and the VAR= argument identifies the input column. Use the ATTRVAR= argument to assign a name that can be used to reference the named association in plot statements within the template.
Reference the attribute map where needed. See “Referencing a Range Attribute Map in Plot Statements” on page 1422.

The RANGEATTRMAP statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in another RANGEATTRMAP statement or in a RANGEATTRMAP statement. The RANGEATTRMAP block must contain at least one RANGE statement. All values that are referenced in the attribute map must be numeric. Data values that are not accounted for in the attribute map receive the default color from the GraphOther, GraphOverflow, or GraphUnderflow style element, depending on where the unassigned values are relative to the specified data ranges.

Note: Unlike a DISCRETEATTRMAP statement, a RANGEATTRMAP statement cannot be directly referenced in a legend. This is because the RANGE statements can specify keywords like MIN and MAX that require a data association for interpretation.

About the RANGE Statement

A RANGE statement within the RANGEATTRMAP block matches color to a numeric value, or a color ramp to a numeric range. To match colors to multiple values or ranges, specify multiple RANGE statements using the following general syntax for each of the statements:

```
RANGE range-spec <options>;
```

For a specific example, see the “Example Program” on page 1446.

When defining multiple RANGE statements, be careful not to define conflicting ranges. A RANGE statement might be syntactically correct by itself while conflicting with settings in another RANGE statement.

Specifying Color in a RANGE Option

For specifying a color in one of the RANGE options, the color keywords can be any of the following:

- a SAS color name (for example, blue)
- an RGB name (for example, CX0000FF or x0000FF)
- an HLS value (for example, H14E162D)
- a gray-scale color code (for example, GRAYBB)
- an HTML color name (for example, AZURE)
- a SAS session color (for example, DMSBLUE).

Examples:

Example 1: Using a Range Attribute Map to Control Marker Colors

The following graph was generated by the “Example Program” on page 1446. The example defines colors to associate with data ranges in an input column that is used in a grouped scatter plot. The RANGEATTRMAP statement starts the attribute map definition and assigns a name to it. The RANGE statements define the value ranges and the colors to associate with those ranges. Because the SCATTERPLOT statement uses the ContrastColor style attribute for the marker colors in a grouped
plot, the `RANGEALTCOLOR=` and `RANGEALTCOLORMODEL=` options are used in the `RANGE` statement to define the range colors. These options can also be used to define range colors for lines and some text elements.

The highest range value specified, which is 0.002 in this example, does not have to be an actual value in the data. The `RANGEATTRVAR` statement associates the attribute map with the data column Density and assigns the name RANGEVAR to the association. The `SCATTERPLOT` statement references the named association in its `MARKERCOLORGRADIENT=` option.

Example Program

```plaintext
proc template;
define statgraph attrmap;
begingroup;
    entrytitle "Height and Weight Distribution" ;

    /* Define the attribute map and assign the name "densityrange." */
    rangeattrmap name="densityrange" ;
        range MIN - 0.0004 / rangealtcolor=blue ;
        range 0.0004 < - 0.0014 / rangealtcolormodel=(lightpurple lightred) ;
        range 0.0014 < - 0.002  / rangealtcolor=red ;
    endrangeattrmap ;

    /* Associate the attribute map with input data column Density and assign variable name RANGEVAR to the named association */
    rangeattrvar attrvar=rangevar var=density attrmap="densityrange" ;

    /* Reference RANGEVAR in the SCATTERPLOT statement */
    layout overlay;
        scatterplot x=height y=weight / markercolorgradient=rangevar
```

![Height and Weight Distribution](image-url)
Example 2: Using a Range Attribute Map to Control Fill Colors

The following graph was generated by the "Example Program" on page 1447. The example defines colors to associate with data ranges in an input column that is used in a surface plot. The RANGEATTRMAP statement starts the attribute map definition and assigns a name to it. The RANGE statements define the value ranges and the colors to associate with those ranges. Because the SURFACEPLOTPARM statement uses the Color style attribute for fill colors, the RANGECOLORMODEL= option is used in the RANGE statement to define the range colors.

The RANGEATTRVAR statement associates the attribute map with the data column Depth and assigns the name DEPTHRANGE to the association. The SURFACEPLOTPARM statement references the named association in its COLORRESPONSE= option.

Example Program

```plaintext
proc template;
  define statgraph surfaceplot;
    dynamic type;
    begingraph;
      /* Define the attribute map and assign the name "depth." */
      rangeattrmap name='depth';
```
range -10 - MAX / rangecolormodel=(green white);
range -20 - -10 / rangecolormodel=(blue green);
range MIN - -20 / rangecolormodel=(darkred blue);
endrangeattrmap;

/* Associate the attribute map with input data column Depth and
assign variable name DEPTHRANGE to the named association. */
rangeattrvar attrvar=depthrange var=Depth attrmap='depth';
entrytitle "Surface Plot of Lake Bed";
layout overlay3d /
   xaxisopts=(griddisplay=on)
   yaxisopts=(griddisplay=on)
   zaxisopts=(griddisplay=on);

/* Reference DEPTHRANGE in the SURFACEPLOTPARM statement */
surfaceplotparm x=length y=width z=depth / name="depth"
   reversecolormodel=true
   colorresponse=depthrange;
continuouslegend "depth";
endlayout;
endgraph;
end;

PROC G3GRID DATA=SASHHELP.LAKE OUT=GRIDDED;
   GRID WIDTH*LENGTH = DEPTH / NAXIS1=75 NAXIS2=75;
RUN;
PROC SGRENDER DATA= GRIDDED TEMPLATE=SURFACEPLOT;
RUN;

RANGEATTRVAR Statement

Creates a named association between a range attribute map of numeric values or value ranges and an
input data column.

Requirements: You must place the RANGEATTRVAR statement directly inside the BEGINGRAPH block
between the BEGINGRAPH statement and the first LAYOUT statement. Do not embed it
in any other GTL statement block.

A range attribute map must be created using the RANGEATTRMAP statement.

See: “Example 1: Using a Range Attribute Map to Control Marker Colors” on page 1445 for an
e xample.

Syntax

RANGEATTRVAR ATTRVAR=attrvar-name
VAR=data-column | expression | dynamic
ATTRMAP="attrmap-name";
Required Arguments

**ATTRVAR=attrvar-name**

specifies a SAS name for the map. This name must be unique among the RANGEATTRVAR statements within the template. The assigned name can be referenced by other statements that can be associated with the range attribute map. the attrvar-name

Restriction Using an attribute variable in an expression is not supported.

**VAR=data-column | expression | dynamic**

specifies a numeric input data column to be associated with an attribute map at run time. If an expression is used, then a new column of transformed values is created and then matched with the attribute map.

Interaction If the column is not found or the column is of the wrong type for the attribute map, then the RANGEATTRVAR statement is ignored.

**ATTRMAP="attrmap-name"**

specifies the name of an existing range attribute map.

Restriction The attrmap-name is case sensitive, must be enclosed in quotation marks, and must be the name that was assigned to the attribute map in the RANGEATTRMAP statement’s NAME= argument.

Details

The RANGEATTRVAR statement creates and names an association between colors that are specified in a RANGEATTRMAP block and a numeric column that is in the data. Attribute maps can be useful for controlling the application of gradient color in a graph or specifying data values that are independent of the actual data. The name that is assigned to the association in the RANGEATTRVAR statement is the name that plot statements must reference to use the attribute map.

Defining and using a numeric-range attribute map requires you to coordinate settings on several statements. For more information, see the RANGEATTRMAP statement’s “Creating and Using a Range Attribute Map” on page 1444.

The RANGEATTRVAR statement must be located within the BEGINGRAPH block but outside of the outermost layout block. It cannot be nested in a RANGEATTRMAP statement.
Dynamic Variables and Macro Variables

Template Types on PROC TEMPLATE

PROC TEMPLATE supports different template types such as COLUMN, TABLE, HEADER, FOOTER, and STATGRAPH. All of these template types support run-time variable substitution via dynamic variables or macro variables. For STATGRAPH templates such variables should be declared within the scope of the template definition before the BEGINGRAPH block.

PROC TEMPLATE;
  DEFINE STATGRAPH template-name ;
  DYNAMIC variable-1"text-1" <... variable-n"text-n">;
  MVAR variable-1"text-1" <... variable-n"text-n">;
  NMVAR variable-1"text-1" <... variable-n"text-n">;
  NOTES "text";

  BEGINGRAPH;
    GTL statements;
    ENDDATA;
  END;
RUN;

DYNAMIC, MVAR, and NMVAR Statements

Each of the DYNAMIC, MVAR, and NMVAR statements can define multiple variables and an optional text-string denoting its purpose or usage. For example:
DYNAMIC YVAR "required" YLABEL "optional";
MVAR LOCATE "INSIDE or OUTSIDE" SYSDATE;
NMVAR TRANS "transparency factor";

Note: For template readability, it is helpful to adopt a naming convention for these
variables to distinguish them from actual option values or column names. Common
conventions include capitalization, or adding leading or trailing underscores to their
names.

Dynamic variables and macro variables can be referenced within the template
definition as

- argument or option values. For example:

  seriesplot x=date y=YVAR / curvelabel=YLABEL
curvelabellocation=LOCATE datatransparency=TRANS;

- parts of some text strings. For example:

  entrytitle "Time Series for " YLABEL;
  entryfootnote "Created on " SYSDATE;

Dynamic variables and run-time macro variable references cannot resolve to
statement or option keywords.

Note that macro variable references should not be prefaced with an ampersand (&)
if you want them to resolve at run time.

Macro variables defined by MVAR are strings when they resolve, as with SYMGET() in
the DATA step.

Macro variables defined by NMVAR are converted to numeric tokens when they
resolve, as with SYMGETN() in the DATA step.

The values for a dynamic variable do not have to be provided by the data source.
Rather, you can provide the values in the DYNAMIC statement in PROC
SGRENDER, specifying the values as a space delimited list, enclosed in quotation
marks. Do not use parentheses in the specification.

In the following example, the graph template specifies a dynamic variable named
TICKS, which is referenced on the XAXISOPTS= option in LAYOUT OVERLAY. The
DYNAMIC statement in PROC SGRENDER provides values for TICKS:

```plaintext
proc template;
  define statgraph regress;
    dynamic TICKS ;
  begingraph;
    layout overlay /xaxisopts=(linearopts={(tickvaluelist=TICKS)});
      scatterplot x=age y=weight;
    endlayout;
  endgraph;
end;
run;

proc sgrender data=sashelp.class template=regress;
  dynamic TICKS="11 13 16" ;
run;
```
If your template uses a dynamic variable to specify a required attribute, such as a variable name, and the name is misspelled or is not provided in the SGRENDER procedure, then a warning is issued and the respective plot statement drops out of the final graph. A graph is produced, but it might be a blank graph, or it might show the results of all statements except those that are in error.

For more information about using dynamic variables and macro variables in your templates, see *SAS Graph Template Language: User’s Guide*.

---

**Dynamic Variables Compared to Macro Variables**

The main difference between dynamic variables and macro variables is how they are initialized.

For dynamic variables, use the DYNAMIC statement with PROC SGRENDER. For example,

```sas
proc sgrender data=sashelp.class template=timeseries;
  dynamic yvar="inflation" ylabel="Inflation Rate";
run;
```

Values for dynamic variables that resolve to column names or strings should be quoted. Numeric values should not be quoted.

For macro variables, use the current symbol table (local or global) to look up the macro variable values at run time. For example,

```sas
%let locate=inside;
%let trans=0.3;

proc sgrender data=sashelp.class template=timeseries;
  dynamic yvar="inflation" ylabel="Inflation Rate";
run;
```

No initialization is needed for system macro variables like SYSDATE.
Overview

In Graph Template Language (GTL), as in Base SAS, an expression is an arithmetic or logical expression that consists of a sequence of operators, operands, and functions. An operand is a dynamic, a macro variable, a column, a function, or a constant. An operator is a symbol that requests a comparison, logical operation, or arithmetic calculation. In GTL, the expression must be enclosed in an EVAL function.

Expressions can be used to set the following types of option values:

- a constant
- a column
- part of the text for ENTRYTITLE, ENTRYFOOTNOTE, and ENTRY statements

Here is an example of computing constants:

/* create reference lines at computed positions */
referenceline y=eval(mean(height)+2*std(height)) /
   curvelabel="+2 STD";
referenceline y=eval(mean(height));
referenceline y=eval(mean(height)-2*std(height)) /
   curvelabel="-2 STD";

Here is an example of creating a new column:

/* create a new column as a log transformation */
scatterplot x=date y=eval(log10(amount));

Here is an example of building a footnote text string:

/* create a date and time stamp as a footnote */
entryfootnote eval(put(today(),date9.)||" : 
   put(time(),timeampm8.));
When you are building a text string, you can use the `||` operator to concatenate the substrings if none of the substrings are a numeric character value. If one or more of the substrings are a numeric character value such as "1.5", then the `||` operator might produce unexpected results. In that case, use the CATS function to concatenate the strings instead. Here is an example of using the CATS function to build a footnote text string that includes the value of dynamic variable `maxWeight`, which stores a numeric character value:

```plaintext
/* indicate the maximum weight in a footnote */
entryfootnote eval(cats("Maximum weight: ", maxWeight," lbs"));
```

### GTL Expressions Compared to WHERE Expressions

Valid GTL expressions are identical to valid WHERE expressions. See the WHERE statement documentation in Base SAS for a comprehensive list of operators and operands. However, GTL expressions do not perform subset operations as WHERE expressions do. The major difference in the result of a logical GTL expression on a column is that a Boolean value is returned for each observation without changing the number of observations.

For example, in the following line of code, the expression for the Y= argument does not reduce the number of observations plotted.

```plaintext
scatterplot x=name y=eval(height between 40 and 60);
```

Instead, the computed numeric column for the Y= argument consists of 0s and 1s, based on whether each observation’s Height column value is between 40 and 60. Whenever expressions are used to create new columns, a new column name is internally manufactured so that it does not collide with other columns in use.

### An Expression in Statement Syntax

Throughout GTL documentation, you see `expression` used in statement documentation:

- **BOXPLOT**
  - `X=column | expression`
  - `Y=numeric-column | expression < /options>;

For the X= argument, `expression` means any EVAL(expression) that results in either a numeric or character column. An expression that yields a constant is not valid because the X= argument does not accept constants.

Similarly, for the Y= argument, expression means any EVAL(expression) that results in a numeric column. The expression cannot result in a character column or any constant because the Y= argument accepts only a numeric column.

On the following REFERENCELINE statement, the X= argument can be a constant (single line) or a column (multiple lines) that has the same data type as the axis.
This means that EVAL(*expression*) can result in a numeric or character column or constant that agrees with the axis type.

**REFERENCE LINE**

\[ X = \text{x-axis-value} \mid \text{column} \mid \text{expression} \langle\text{options}\rangle; \]

**Automatic Type Conversion.** Although expressions that are used in a DATA step perform automatic type conversion, GTL expression evaluation does not. Thus, you must use function(s) to perform required type conversions in an expression. Otherwise, the expression generates an error condition without warning when the template is executed.

For example, consider the following GTL expression:

\[
\text{if} (\text{substr(value, 1, 2)} = \text{"11"})
\]

This expression uses the SUBSTR function to determine whether the first two characters from VALUE evaluate to the string value "11". If VALUE is a string, then the expression works properly. However, if VALUE is numeric, then the expression generates an error condition. For a numeric, you must convert the value to a string before passing it to the SUBSTR function. The following modification uses the CATS function to perform the type conversion when necessary:

\[
\text{if} (\text{substr(cats(value, 1, 2)}) = \text{"11"})
\]
Overview

GTL supports a large number of functions, including:

- SAS functions that can be used in the context of a WHERE expression
- functions that are defined only in GTL
- summary statistic functions

SAS Functions

SAS Functions That Can Be Used in a GTL Template

Most of the SAS functions that are available in WHERE expressions can be used in a GTL template. These SAS functions include:

- character-handling functions
- date and time functions
mathematical and statistical functions

Not all SAS functions are available in WHERE expressions. Call routines and other DATA-step-only functions (for example, LAG, VNAME, OPEN) are some examples of functions that cannot be used. Not all functions that are available in WHERE expressions are supported in GTL templates in all cases. The following form of the PUT function is an example:

\[
\text{markercharacter} = \text{eval}(\text{put(amount, dollar7.2 -L)})
\]

This form results in an error when the template is compiled. However, the following form is supported.

\[
\text{markercharacter} = \text{eval}(\text{put(amount, dollar7.2)})
\]

If you want to justify a string that is generated by the PUT function, use the LEFT or RIGHT function with the PUT function as shown in the following example:

\[
\text{markercharacter} = \text{eval(\text{left(put(amount, dollar7.2))}}
\]

Functions that accept null parameter values also might not be supported when you specify a null parameter value.

For more information about SAS functions, see "Dictionary of Functions and CALL Routines" in SAS Functions and CALL Routines: Reference.

---

**SAS Functions That Can Be Used to Create Flexible Templates**

The following table shows some of the SAS functions can be used to increase the flexibility of your template code.

<table>
<thead>
<tr>
<th>SAS Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFC(logical-expression, &quot;true-value&quot;, &quot;false-value&quot; &lt;,&quot;missing-value&quot;&gt;)</td>
<td>Returns the character value true-value if logical-expression resolves to TRUE, false-value if it resolves to FALSE, or missing-value if it resolves to a missing value. The TRUE, FALSE, and MISSING values must be enclosed in quotation marks.</td>
</tr>
<tr>
<td>IFN(logical-expression, true-value, false-value &lt;,missing-value&gt;)</td>
<td>Returns the numeric value true-value if logical-expression resolves to TRUE, false-value if it resolves to FALSE, or missing-value if it resolves to a missing value.</td>
</tr>
</tbody>
</table>

---

**Examples of Using the IFC and IFN SAS Functions**

The IFC and IFN functions return one of two character or numeric values based on whether a conditional expression resolves to TRUE or FALSE. They can also return
an optional third value if the conditional expression resolves to a missing value. These functions enable you to specify a value based on a conditional expression, effectively creating a new data column. In some cases, these functions can be used in place of IF-THEN-ELSE statements in your template code. As with other functions, you must enclose the IFC and IFN functions in the EVAL function.

Here is an example that uses both the IFN and IFC functions for creating a sales-based commission chart for employees in a sales group. Each employee in the group works in one of two sales units: Products and Services. The data for this example includes the employee ID, total sales, and sales unit code for each member of the sales group. Here is the data.

```
data sales;
  input empID totalSales salesUnit $18;
  format totalSales dollar9.;
datalines;
112876 129489.44 P
112421 169842.97 S
115331 108763.51 S
110765 181009.22 P
113722 147688.78 P
;
```

The TotalSales column contains the total sales for each employee. The SalesUnit column contains a code that identifies the sales unit in which each employee works. The codes are P for the Products unit and S for the Services unit.

Here is the output for this example.

![Sales-Based Commission](image)

The two bar charts show the total sales and earned commission for each employee. The IFN function is used to compute commission for each employee based on his or her total sales. Employees that achieved a sales total of $120,000 or more earn a commission of 5% of their total sales. All other employees earn a commission of 2.5% of their total sales.
An axis table along the X axis shows the sales unit for each employee. The IFC function is used to convert the P and S SalesUnit codes into more descriptive values in the axis table. Because only two sales unit codes are used in this case, the IFC function can be used for this purpose. This eliminates the need to add a new column to the data in a DATA step or to create and apply a custom format to the SalesUnit column.

Here is the SAS code that defines the template and generates the graph.

```sas
proc template;
define statgraph commission;
begingraph;
  entrytitle "Sales-Based Commission";
  layout overlay /
    xaxisopts=(label="Employee ID")
    yaxisopts=(label="Total Sales")
    y2axisopts=(label="Commission"
      linearopts=(viewmax=15000 tickvalueformat=dollar9.));
  /* Generate the sales bar chart. */
  barchart category=empID response=totalSales /
    name="Sales" legendlabel="Total Sales" barwidth=0.3
    discreteoffset=-0.2 fillattrs=graphData1;

  /* Generate the commission bar chart. */
  barchart category=empID
    /* Use IFN to compute the commission. */
    response=eval(ifn(totalSales >= 120000,
      totalSales * 0.05, /* 5% if TRUE */
      totalSales * 0.025)) / /* 2.5% if FALSE */
    name="Commission" legendlabel="Commission"
    barwidth=0.3 yaxis=y2 discreteoffset=0.2
    fillattrs=graphData2;

  /* Add an axis table that shows the sales unit for each employee. */
  innermargin / align=bottom;
  axistable x=empID
    /* Use IFC to convert the codes to meaningful values. */
    value=eval(ifc(salesUnit = 'P','Products",
      "Services")) / display=(values);
  endinnermargin;
  discretelegend "Sales" "Commission";
endlayout;
endgraph;
end;
run;

proc sgrender data=sales template=commission;
run;
```
The following table shows some functions that are used only in GTL. As with other functions, these must be enclosed within an EVAL. In all these functions, column can be either the name of a column in the input data set or a dynamic / macro variable that resolves to such a column.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLC(&quot;string-l&quot;, &quot;string-l&quot;&lt;, &quot;string-n&quot; ...)</td>
<td>Converts a list of comma-separated string values into a temporary character column. Starting with SAS 9.4M2, you can use this function to specify values in options that accept a character column.</td>
</tr>
<tr>
<td>COLN(n-1, n-1&lt;, n-N...)</td>
<td>Converts a list of comma-separated numeric values into a temporary numeric column. You can use this function to specify values in options that accept a numeric column.</td>
</tr>
<tr>
<td>COLNAME(column)</td>
<td>Returns the case-sensitive name of the column.</td>
</tr>
<tr>
<td>COLLABEL(column)</td>
<td>Returns the case-sensitive label of the column. If no label is defined for the column, then the case-sensitive name of the column is returned.</td>
</tr>
<tr>
<td>EXISTS(item)</td>
<td>Returns 1 if specified item exists, 0 otherwise. If item is a column, then it tests for the presence of the column in the input data set. If item is a dynamic / macro variable, then it tests whether there has been a run-time initialization of the variable.</td>
</tr>
<tr>
<td>EXPAND(numeric-column, freq-column)</td>
<td>Creates a new column as (numeric-column * frequency-column).</td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ASORT(<code>column</code>, RETAIN=ALL)</td>
<td>Sorts all columns of the data object by the values of <code>column</code> in ascending order. <strong>Warning:</strong> if the RETAIN=ALL argument is not included, <code>column</code> alone is sorted, not the other columns, causing rowwise information to be lost. <strong>Limitation:</strong> only one sort operation (whether an ASORT() or DSORT() function) can be used within a single template definition.</td>
</tr>
<tr>
<td>DSORT(<code>column</code>, RETAIN=ALL)</td>
<td>Sorts all columns of the data object by the values of <code>column</code> in descending order. <strong>Warning:</strong> if the RETAIN=ALL argument is not included, <code>column</code> alone is sorted, not the other columns, causing rowwise information to be lost. <strong>Limitation:</strong> only one sort operation (whether an ASORT() or DSORT() function) can be used within a single template definition.</td>
</tr>
<tr>
<td>NUMERATE(<code>column</code>)</td>
<td>Returns a <code>column</code> that contains the ordinal position of each observation in the input data set (similar to an Obs column).</td>
</tr>
</tbody>
</table>

**Examples**

```sas
/* arrange bars in descending order of response values */
barchartparm category=region response=eval(dsord(amount,retain=all));

/* label outliers with their position in the data set */
/* it does not matter which column is used for NUMERATE() */
boxplot x=age y=weight / datalabel=eval(numerate(age));

/* add information about the column being processed, which is passed by a dynamic */
entrytitle "Distribution for " eval(colname(DYNVAR));
```

**Using the TYPEOF SAS Function**

The TYPEOF function returns the type of a specified column at run time.
TYPEOF(column)

This function returns the character ‘C’ if the specified column is a character column or ‘N’ if it is a numeric column.

You can use the TYPEOF function to take specific actions in your template at runtime based on the input data type. Here is an example that creates a graph of two columns and uses the TYPEOF function to select a graph type that is appropriate for the column types. The result returned by the TYPEOF function determines the graph type as follows:

- If both columns are numeric, then it creates a scatter plot.
- If the X column is character and the Y column is numeric, then it creates a vertical bar chart.
- If the X column is numeric and the Y column is character, then it swaps the category and response columns in the BARCHART statement and orients the chart horizontally.

Here is the output for the third case, a numeric X column and a character Y column.

Here is the SAS code.

```sas
/* Define the graph template. */
proc template;
    define statgraph plot;
        dynamic cat resp; /* Category and response columns. */
        begingraph;
            entrytitle "Graph of " eval(collabel(resp)) " and "
                   eval(collabel(cat));
            layout overlay;
            /* If cat and resp are numeric, then generate a scatter plot. 
               Otherwise, generate a bar chart. */
            if (typeof(cat) = "N" and typeof(resp) = "N")
                scatterplot x=cat y=resp;
            else
                /* If cat is a character column, then generate a vertical bar chart. Otherwise, generate a horizontal bar chart. */
                if (typeof(cat) = "C")
```
GTL Summary Statistic Functions

Commonly Used Summary Statistic Functions

Several GTL summary statistic functions are available that return a numeric constant, based on a summary operation on a numeric column. The results are the same as if the corresponding statistics were requested with PROC SUMMARY. These functions take a single argument that resolves to the name of a numeric column. Here is an example.

\[
\text{number} = \text{EVAL}\left(\text{function-name}(\text{numeric-column})\right)
\]

The GTL summary statistic functions take precedence over similar multi-argument DATA step functions. The following table lists the GTL summary statistic functions that you can use.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSS</td>
<td>Corrected sum of squares</td>
</tr>
<tr>
<td>CV</td>
<td>Coefficient of variation</td>
</tr>
<tr>
<td>KURTOSIS</td>
<td>Kurtosis</td>
</tr>
<tr>
<td>LCLM</td>
<td>One-sided confidence limit below the mean</td>
</tr>
<tr>
<td>MAX</td>
<td>Largest (maximum) value</td>
</tr>
<tr>
<td>Function Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>MEAN</td>
<td>Mean</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>Median (50th percentile)</td>
</tr>
<tr>
<td>MIN</td>
<td>Smallest (minimum) value</td>
</tr>
<tr>
<td>N</td>
<td>Number of nonmissing values</td>
</tr>
<tr>
<td>NMISS</td>
<td>Number of missing values</td>
</tr>
<tr>
<td>P1</td>
<td>1st percentile</td>
</tr>
<tr>
<td>P5</td>
<td>5th percentile</td>
</tr>
<tr>
<td>P25</td>
<td>25th percentile</td>
</tr>
<tr>
<td>P50</td>
<td>50th percentile</td>
</tr>
<tr>
<td>P75</td>
<td>75th percentile</td>
</tr>
<tr>
<td>P90</td>
<td>90th percentile</td>
</tr>
<tr>
<td>P95</td>
<td>95th percentile</td>
</tr>
<tr>
<td>P99</td>
<td>99th percentile</td>
</tr>
<tr>
<td>PROBT</td>
<td>p-value for Student’s t statistic</td>
</tr>
<tr>
<td>Q1</td>
<td>First quartile</td>
</tr>
<tr>
<td>Q3</td>
<td>Third quartile</td>
</tr>
<tr>
<td>QRANGE</td>
<td>Interquartile range</td>
</tr>
<tr>
<td>RANGE</td>
<td>Range</td>
</tr>
<tr>
<td>SKEWNESS</td>
<td>Skewness</td>
</tr>
<tr>
<td>STDDEV</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>STDERR</td>
<td>Standard error of the mean</td>
</tr>
<tr>
<td>SUM</td>
<td>Sum</td>
</tr>
<tr>
<td>SUMWGT</td>
<td>Sum of weights</td>
</tr>
<tr>
<td>T</td>
<td>Student’s t statistic</td>
</tr>
</tbody>
</table>
### Example

The following example uses GTL summary statistic functions to dynamically construct reference lines and a table of statistics for a numeric variable, which is supplied at run time.

Here is the graph for this example.

![Distribution of MRW](image)

Here is the SAS code.
```
proc template;
    define statgraph expression;
        dynamic NUMVAR "required";
    begingraph;
        entrytitle "Distribution of " eval(colname(NUMVAR));
        layout overlay / xaxisopts=(display=(ticks tickvalues line));
        histogram NUMVAR;

        /* create reference lines at computed positions */
        referenceline x=eval(mean(NUMVAR)+2*std(NUMVAR)) /
            lineattrs={pattern=dash} curvelabel="+2 STD";
```

/* create inset */
layout gridded / columns=2 order=rowmajor
  autoalign=(topleft topright) border=true;
  entry halign=left "N";
  entry halign=left eval(strip(put(n(NUMVAR),12.0)));
  entry halign=left "Mean";
  entry halign=left eval(strip(put(mean(NUMVAR),12.2)));
  entry halign=left "Std Dev";
  entry halign=left eval(strip(put(stddev(NUMVAR),12.2)));
endlayout;
endlayout;
endgraph;
end;
run;

proc sgrender data=sashelp.heart template=expression;
  dynamic numvar="MRW";
run;
Overview

GTL supports conditional logic that enables you to include or exclude one or more GTL statements at run time:

\[
\text{IF (condition)} \\
\quad \text{GTL-statement(s);} \\
\text{ELSE} \\
\quad \text{GTL-statement(s);} \\
\text{ENDIF;}
\]

The IF statement requires an ENDIF statement. The IF block can be placed anywhere within the BEGINGRAPH / ENDGRAPH block.

The condition is an expression that evaluates to a numeric constant, where all numeric constants other than 0 and MISSING are true. There is an implied EVAL(condition), so it is not necessary to include an EVAL as part of the condition.

Examples:

/* test a computed value */
if (weekday(today()) in (1 7))
  entrytitle "Run during the work week";
else
  entrytitle "Run during the weekend";
endif;

/* test for the value a numeric dynamic */
if (ADDREF > 0)
  referenceline y=1;
  referenceline y=0;
  referenceline y=-1;
endif;

/* test for the value a character dynamic */
if (upcase(ADDREF) =: "Y")
Conditional Logic Determines Statement Rendering

The GTL conditional logic is used only to determine which statements are rendered, not to control what is in the data object. In the following example, the data object contains columns for Date, Amount, and LOG(AMOUNT), but only one scatter plot is created.

```g Nähe
if (LOGFLAG)
  scatterplot x=date y=amount;
else
  scatterplot x=date y=log(amount);
endif;
```

Also, it is seldom necessary to test for the existence of option values set by columns or dynamic variables. Consider the following statement:

```g Nähe
scatterplot x=date y=amount / group=GROUPVAR;
```

This SCATTERPLOT statement is equivalent to the following code because option values that are set by columns that do not exist or dynamic variables that are uninitialized simply “drop out” at run time and do not produce errors or warnings:

```g Nähe
if (exists(GROUPVAR))
  scatterplot x=date y=amount / group=GROUPVAR;
else
  scatterplot x=date y=amount;
endif;
```

The GTL code that is conditional must be complete statements, or complete blocks of statements, or both. The following IF block produces a compile error because there are more LAYOUT statements than ENDLAYOUT statements:

```g Nähe
/* this IF block produces a compile error */
if (exists(SQUAREPLOT))
  layout overlayequated / equatetype=square;
else
  layout overlay;
```
This is the correct conditional construct:

```glossary
if (exists(SQUAREPLOT))
  layout overlayequated / equatetype=square;
  scatterplot x=XVAR y=YVAR;
  endlayout;
else
  layout overlay;
  scatterplot x=XVAR y=YVAR;
  endlayout;
endif;
```

GTL Does Not Provide ELSE IF Syntax

GTL does not provide ELSE IF syntax, but you can create a nested IF/ELSE block as follows:

```glossary
IF (condition-1)
  GTL-statement(s);
ELSE
  IF (condition-2)
    GTL-statement(s);
  ELSE
    GTL-statement(s);
  ENDIF;
ENDIF;
```
PART 13

Appendixes

Appendix 1
Syntax Conventions and Argument Value Types ........................................ 1479

Appendix 2
ODS GRAPHICS Statement ............................................................................ 1483

Appendix 3
Reserved Keywords and Unicode Values ...................................................... 1515

Appendix 4
Display Attributes ......................................................................................... 1519

Appendix 5
SAS Formats Not Supported ........................................................................ 1541

Appendix 6
Generalized Macro for BOXPLOTPARM Data ............................................. 1545

Appendix 7
Memory Management for ODS Graphics ..................................................... 1551
Syntax Conventions

< > ==> optional
a | b ==> either a or b

Value Types for Statement Options

The default value of an option depends on the template definition that is in use, and the implementation of that option depends on the ODS destination that formats the output. In addition, if you are creating HTML output, then the implementation of an attribute depends on the browser that you use.

This section describes the value types that are available for GTL statement options.

*boolean*

specifies a literal value that resolves to true or false. The following table lists literal values that resolve to true or false.

<table>
<thead>
<tr>
<th>Values That Resolve To True</th>
<th>Values That Resolve To False</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td><em>ON</em></td>
<td><em>OFF</em></td>
</tr>
<tr>
<td>TRUE</td>
<td>FALSE</td>
</tr>
<tr>
<td>Values That Resolve To True</td>
<td>Values That Resolve To False</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td><em>YES</em></td>
<td><em>NO</em></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

`color` specifies a string that identifies a color. A color can be one of the following:

- any of the color names that are supported by SAS. See "Color-Naming Schemes" on page 1528.
- one of the colors that exists in the SAS session when the style template is used, such as DMSBLACK or DMSCYAN. (Use these color specifications only if you are running SAS in the windowing environment.)
- an English description of an Hue/Light/Saturation (HLS) value. Such descriptions use a combination of words to describe the lightness, the saturation, and the hue (in that order). You can use the Color Naming System to form a color by doing one of the following:
  - combining a chromatic hue with a lightness, a saturation, or both
  - combining the achromatic hue gray with a lightness
  - combining the achromatic hue black or white without qualifiers.
  - combining words to form a wide variety of colors, such as light vivid green, dark vivid orange, or light yellow.
- specify hues that are intermediate between two neighboring colors. To do so, combine one of the following adjectives with one of its neighboring colors: brownish, greenish, purplish, or yellowish (for example, bluish purple or reddish orange).

`column` specifies a column variable that contains either double-precision values or string values, or a dynamic variable that refers to such a column.

See also: `integer-column`, `numeric-column`, and `string-column`.

`dimension` specifies a nonnegative number. The number can be followed by one of the following optional units of measure:

**Table A12.2 Units for Dimension**

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM</td>
<td>centimeters</td>
</tr>
<tr>
<td>IN</td>
<td>inches</td>
</tr>
<tr>
<td>MM</td>
<td>millimeters</td>
</tr>
<tr>
<td>PCT or %</td>
<td>percentage</td>
</tr>
</tbody>
</table>
**expression**

specifies a selective, relational, or logical program structure that calculates values when those values are not stored in the data. The expression must be specified as an EVAL() argument. The following shows the structure of an EVAL() argument:

\[ x = \text{EVAL(expression)} \]

The expression returns a number and can be formed with consonants, data columns, dynamic variables, functions, or other expressions. The following example uses the data column Time and the SGE functions MEAN and ACF:

\[ \text{EVAL(MEAN(Time) + ACF(Time, NLags=10))} \]

For more information about expressions, see Chapter 21, “Expressions,” on page 1457.

**fill-pattern**

specifies a fill pattern as a two-character code that consists of a line-direction prefix (R for right, L for left, or X for cross hatch) and a line identification number, 1–5. For more information about fill patterns, see “Fill Pattern Options” on page 1521.

**format**

specifies a SAS format or a user-defined format.

**integer, integer-column**

specifies a member of the set of positive whole numbers, negative whole numbers, and zero.

An integer column specifies a column that contains integer values, or a dynamic variable that refers to such a column.

**line-pattern-name, line-pattern-number**

specifies a string value of a line pattern, a numeric value of a line pattern, a dynamic variable that contains such a string or number, or a style reference to a line pattern. Line patterns are chosen for discriminability. Because of different densities, equal weighting is impossible for lines of the same thickness. Instead, line patterns are ordered to provide a continuum of weights, which is useful when displaying confidence bands.

For details about line attributes, see “Line Options” on page 1522.

**marker-name**

specifies a string value of a maker symbol, a dynamic variable that contains a marker symbol, or a style reference to a marker symbol.

For details about marker attributes, see “Marker Options” on page 1523.

**number, numeric-column**

specifies a value, a dynamic variable that contains a double-precision value, an expression that resolves to a double-precision value, or a style reference to a double-precision value.
A numeric-column specifies a column that contains double-precision values, or a dynamic variable that refers to such a column.

string, string-column
speifies a quoted character string.

A string-column specifies a column that contains string values, or a dynamic variable that refers to such a column.

---

Note: For quoted character string options in GTL, a space enclosed in quotation marks (" " or ' ') and empty quotation marks ("" or "") are not equivalent. A space enclosed in quotation marks specifies a blank space or a missing string value. Empty quotation marks have the same effect as not setting the option. To specify a blank space or missing value in a quoted string option, use a space enclosed in quotation marks (" " or ' ').

---

style-reference
specifies a reference to an attribute that is defined in a style element.

In the ODS Graphics templates that SAS provides, options for plot features are specified with a style reference in the form style-element-name:attribute-name, rather than a specific value. For example, the symbol, color, and size of markers for a basic scatter plot is specified in a SCATTERPLOT statement as follows:

```plaintext
scatterplot x=X y=Y /
   markersymbol=GraphDataDefault:markersymbol
   markercolor=GraphDataDefault:contrastcolor
   markersize=GraphDataDefault:markersize
```

The above style references guarantee a common appearance for markers used in all basic scatter plots. For non-grouped data, the marker appearance is controlled by the GraphDataDefault style element in the style template that you specify.

In order to create your own style template, or to modify a style template to use with ODS Graphics, you need to understand the relationship between style elements and graph features. For more information, see the usage guide.
ODS GRAPHICS Statement

Enables or disables ODS Graphics processing and sets graphics environment options. This statement affects ODS template-based (ODS Graphics) graphics only. The ODS GRAPHICS statement does not affect device-based graphics (SAS/GRAPH).

Valid in: Anywhere
Category: ODS: Output Control
Default: ON. Beginning in SAS 9.4, ODS Graphics is enabled by default on all platforms except z/OS. When running SAS in batch mode, the default is OFF.
Interaction: SAS/GRAPH device-based global statements such as GOPTIONS, SYMBOL, PATTERN, AXIS, and LEGEND do not affect template-based graphics. The ODS GRAPHICS statement does not affect device-based graphics.
See: For information about common tasks for managing ODS Graphics output, see SAS Graph Template Language: User’s Guide.

Syntax

ODS GRAPHICS <OFF | ON> </options> ;

Summary of Optional Arguments

ANTIALIAS | NOANTIALIAS | ANTIALIAS= ON | OFF
specifies whether anti-aliasing is applied to the rendering of the line and markers in any graph.

ANTIALIASMAX= n
specifies the maximum number of graphics elements before anti-aliasing is disabled.

ATTRPRIORITY=COLOR | NONE
specifies a priority for cycling of the group attributes.

BORDER | NOBORDER | BORDER=ON | OFF
specifies whether to draw a border around each graph.

BYLINE=NOBYLINE | TITLE | FOOTNOTE
specifies how the BY line is displayed in graphs.

DATASKINMAX=n
specifies the maximum number of graphical elements allowed per plot when a data skin is applied.

DISCRETEMAX=n
specifies the maximum number of discrete values to be shown in any graph.

DRILLTARGET="_blank" | "_self" | "_parent" | "_top" | "frame-name"
specifies the window that displays the drill-down output.

GROUPMAX=n
specifies the maximum number of group values to be shown in any graph.

HEIGHT=dimension
specifies the height of a graph.

IMAGEMAP | NOIMAGEMAP | IMAGEMAP=ON | OFF
specifies whether data tips are generated.

IMAGENAME="filename"
specifies the base image filename.

LABELMAX=n
specifies the maximum number of labeled areas before data label collision avoidance is disabled.

LABELPLACEMENT= GREEDY | SA
specifies the label-placement algorithm to use for positioning labels in the graphs.

LEGENDAREAMAX=n
specifies an integer that is interpreted as the maximum percentage of the overall graphics area that a legend can occupy.

LINEPATTERNOBSMAX=n
specifies the maximum the number of vertices for a patterned line.

LOESSOBSMAX=n
specifies an upper limit for the number of observations that can be used with a loess plot.

NBINSMAX=n
specifies the maximum number of bins that are processed for histograms.

NXYBINSMAX=n
specifies the maximum number of bins that are processed for heat maps.

OBSMAX=n
specifies the maximum number of observations that are processed.

OUTPUTFMT=file–type | STATIC
specifies the output format used to generate image or vector graphic files.

PANELCELLMAX=n
specifies the maximum number of cells in a graph panel where the number of cells is determined dynamically by classification variables.
pushes and pops ODS GRAPHICS settings in a stack.

**RESET | RESET= option**
Reset one or more ODS GRAPHICS options to its default.

**SCALE | NOSCALE | SCALE=ON | OFF**
specifies whether the content of a graph is scaled proportionally.

**SCALEMARKERS | NOSCALEMARKERS | SCALEMARKERS=ON | OFF**
specifies whether the plot markers are to be scaled with the graph size.

**SHOW**
writes the current ODS Graphics settings to the SAS log.

**STACKDEPTHMAX=n**
specifies the maximum stack depth for PUSH and POP requests.

**SUBPIXEL | NOSUBPIXEL | SUBPIXEL=ON | OFF**
specifies whether subpixel rendering should be used for rendering ODS Graphics.

**TIPMAX=n**
specifies the maximum number of distinct data tip areas allowed before data tips are disabled.

**TOTALCELLMAX=n**
specifies the maximum number of total cells in a graph where the number of cells is determined dynamically by classification variables.

**WIDTH=dimension**
specifies the width of any graph.

**Without Arguments**
If ODS Graphics is currently disabled, then specifying the ODS GRAPHICS statement without options enables it. If ODS Graphics is currently enabled, then specifying the ODS GRAPHICS statement leaves it enabled.

**Required Arguments**

**ON**
enables ODS Graphics processing. This is the default if no argument is used.

Note: Beginning in SAS 9.4, ODS Graphics is enabled by default on all platforms except z/OS.

**Alias**

**YES**

**OFF**
disables ODS Graphics processing.

**Alias**

**NO**

**Optional Arguments**

**ANTIALIAS | NOANTIALIAS | ANTIALIAS=ON | OFF**
specifies whether anti-aliasing is applied to the rendering of the line and markers in any graph. Anti-aliasing smooths the appearance of lines and some markers. Text displayed in the graph is always anti-aliased. For graphical displays that plot large numbers of points it is recommended that ANTIALIAS=OFF be specified for performance considerations.
ANTIALIAS
  smooths jagged edges of all components in the graph.

NOANTIALIAS
  does not smooth jagged edges of components other than text in the graph.

ANTIALIAS=ON | OFF
  specifies whether anti-aliasing is applied to the rendering of the line and
  markers in the graph.

  ON
    smooths jagged edges of all components in the graph.
    Alias YES

  OFF
    does not smooth jagged edges of components other than text in the
    graph.
    Alias NO

Default  ANTIALIAS or ANTIALIAS=ON | YES

Restriction  If the number of markers or lines in the plot exceeds the number
            specified by the ANTIALIASMAX= option, then the ANTIALIAS
            option is disabled. This is true even if you specify the option
            ANTIALIAS=ON or ANTIALIAS.

ANTIALIASMAX= n
  specifies the maximum number of graphics elements before anti-aliasing is
  disabled. For example, if there are more than 400 scatter point markers to be
  anti-aliased and ANTIALIASMAX=400, then no markers are anti-aliased. The
  default value is 4000.

Note: Prior to SAS 9.4M3, the ANTIALIASMAX= option specifies the maximum
number of observations in the graph data to be anti-aliased before anti-aliasing
is disabled. The default is 4000. When the graph data contains more than 4000
observations, anti-aliasing is disabled for the entire graph. Starting with SAS
9.4M3, the ANTIALIASMAX= option specifies the maximum number of graphics
elements to be anti-aliased in each plot on a per-plot basis. The default remains
at 4000. If any plot in a graph contains more than 4000 elements, anti-aliasing is
disabled for that plot. Anti-aliasing is enabled for the rest of the graph in that

n
  specifies a positive integer.

Default  4000

ATTRPRIORITY=COLOR | NONE
  specifies a priority for cycling of the group attributes.

COLOR
  assigns priority to the color attribute rotation by cycling through the list of
  colors while holding the marker symbol and line pattern constant. When all of
  the colors are exhausted, the marker symbol and line style attributes
  increment to the next element, and then the colors in the list are repeated.
  This pattern repeats as needed.
NONE
does not use an attribute priority in the rotation pattern, even if one is set in
the active style’s AttrPriority attribute. The rotation pattern cycles
progressively through the attribute lists.

Default
The AttrPriority attribute of the graph style element, or NONE if the
current style does not define the AttrPriority style attribute.

Interaction
The default lists of data colors, contrast colors, marker symbols, and
line patterns are set in the active style’s GraphData1–GraphDataN
elements.

Tip
Use the ATTRPRIORITY=None option if you want groups to be
distinguished by color, marker, and line changes for all styles that
use color.

BORDER | NOBORDER | BORDER=ON | OFF
specifies whether to draw a border around each graph.

BORDER
draws a border around the graph.

NOBORDER
does not draw a border around the graph.

BORDER=ON | OFF
specifies whether to draw the graph with a border on the outermost layout.

ON
draws a border around the graph.

Alias YES

OFF
does not draw a border around the graph.

Alias NO

Default BORDER or BORDER=ON | YES

BYLINE=NOBYLINE | TITLE | FOOTNOTE
specifies how the BY line is displayed in graphs. The option specifies how the
BY line is displayed when an analysis is run with a BY statement. By default, no
BY line is displayed.

The following code is an example of how the placement of the BY line is
controlled in most graph templates:

if (_BYTITLE_)
  entrytitle _BYLINE_ / textattrs=GraphValueText;
else
  if (_BYFOOTNOTE_)
    entryfootnote halign=left _BYLINE_;
  endif;
endif;
endif;

You can modify the graph template if you want to change how the BY line is
displayed. Because most graphs have titles and few graphs have footnotes, the
BY line looks better when it is displayed as a footnote. For complete
documentation about the Graph Template Language, see SAS Graph Template Language: User’s Guide.

When the BYLINE= option is specified, and there are BY groups, ODS creates a BY line and sets the appropriate special dynamic variables. The following table lists the special dynamic variables for BY lines. For complete documentation about special dynamic variables, see “Special Dynamic Variables” in SAS Graph Template Language: User’s Guide.

**Table A13.1  Special Dynamic Variables for BY Lines**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>BYFOOTNOTE</em></td>
<td>This variable is set to 1 when you specify a BY statement and the ODS GRAPHICS BYLINE= option is set to FOOTNOTE. Otherwise, the variable is set to 0 or is NULL.</td>
</tr>
<tr>
<td><em>BYTITLE</em></td>
<td>This variable is set to 1 when you specify a BY statement and the ODS GRAPHICS BYLINE= option is set to TITLE. Otherwise, the variable is set to 0 or is NULL.</td>
</tr>
<tr>
<td><em>BYLINE</em></td>
<td>This variable contains the text string that can be displayed as a title or footnote.</td>
</tr>
</tbody>
</table>

The variables in the table are set automatically only for analytical procedures that support ODS Graphics. For all other procedures, the variables are not set automatically (NULL). To determine whether the procedure you are using supports ODS Graphics, refer to the procedure documentation.

**NOBYLINE**  
specifies that no BY line is displayed. NOBYLINE is the default.

**FOOTNOTE**  
specifies that the BY line is displayed as a left-justified graph footnote. This is the recommended setting.

**TITLE**  
specifies that the BY line is displayed as a centered graph title. Specifying TITLE is not recommended because graphs are not designed to have additional title lines.

**Default**  
NOBYLINE

**Restriction**  
This option does not work with the ODS Graphics procedures such as SGPLOT and SGPANEL. To remove BY lines in those procedures, use the NOBYLINE SAS system option. Example: 

```sas
options nobyline;
```

**DATASKINMAX=n**  
specifies the maximum number of graphical elements allowed per plot when a data skin is applied.

**Note:** This feature applies to SAS 9.4M1 and later releases.

**n**  
specifies a positive integer.

**Default**  
200
DISCRETEMAX=n
specifies the maximum number of discrete values to be shown in any graph. Bar charts and box plots are examples of affected plot types. Scatter plots and other plot types can be affected if the data to be plotted is discrete or the axis is discrete.

\[n\]
specifies a positive integer.

Default 1000

Tips
Some plot layers might be unaffected by the DISCRETEMAX= option, and those layers are rendered. If all layers are affected, a blank graph is rendered.

If the value specified by the DISCRETEMAX= option is exceeded by any plot layer in the graph, that layer is not drawn and a warning message is issued. In that case, use the DISCRETEMAX= option to increase the maximum number of discrete values that are allowed. Starting with SAS 9.4M5, the log message includes a suggested value for DISCRETEMAX=.

DRILLTARGET="_blank" | "_self" | "_parent" | "_top" | "frame-name"
specifies the window that displays the drill-down output.

Note: This option is supported only for HTML.

"_blank"
opens a new browser window to display the drilldown output.

Default _blank is the default.

Requirements You must enclose _blank in quotation marks.

You must specify _blank in lowercase.

"_self"
opens the drill-down output in the same window.

Requirements You must enclose _self in quotation marks.

You must specify _self in lowercase.

"_parent"
opens the drill-down output in the parent frame.

Requirements You must enclose _parent in quotation marks.

You must specify _parent in lowercase.

"_top"
opens the drill-down output in the full body of the window.

Requirements You must enclose _top in quotation marks.

You must specify _top in lowercase.
"frame-name"
  opens the drill down output in the named frame in the current window. If the
  name does not exist, the output is opened in a new window.

  **Requirement**  You must enclose `frame-name` in quotation marks.

**GROUPMAX=n**
specifies the maximum number of group values to be shown in any graph. Any
graph that supports the `GROUP=` option is affected.

  \[ n \]
  specifies a positive integer.

  **Default**  1000

  **Tip**  If the value specified by the `GROUPMAX=` option is exceeded by any
  plot layer in the graph, that layer is rendered. The system ignores the
  `GROUP=` option and issues a warning message. In that case, use the
  `GROUPMAX=` option to increase the maximum number of group values
  that are allowed. Starting with SAS 9.4M5, the log message includes a
  suggested value for `GROUPMAX=`.

**HEIGHT=dimension**
  specifies the height of a graph.

  \[ dimension \]
  is a nonnegative number followed by one of these units of measure:

  **Table A13.2**  Units of Measure for Dimension

<table>
<thead>
<tr>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>Centimeters</td>
</tr>
<tr>
<td>in</td>
<td>Inches</td>
</tr>
<tr>
<td>mm</td>
<td>Millimeters</td>
</tr>
<tr>
<td>pct</td>
<td>Percentage</td>
</tr>
<tr>
<td>pt</td>
<td>Point size (72 points = 1 inch)</td>
</tr>
<tr>
<td>px</td>
<td>Pixels</td>
</tr>
</tbody>
</table>

  **Defaults**  The value of the SAS registry entry "ODS > ODS GRAPHICS >
  Design Height" or the value of the DesignHeight= option in a
  STATGRAPH template. Typically, the value is 480px.

  For the PRINTER destination, units of 1/150 of an inch are used by default.

  **Tips**  If only the `HEIGHT=` option is specified, then the default aspect ratio of the
  graph is maintained.

  When you change the size of a graph, by default graph elements such as
  markers, lines, text, and titles are scaled proportionally. This scaling takes
To disable the scaling, specify the NOSCALE option.

**IMAGEMAP | NOIMAGEMAP | IMAGEMAP=ON | OFF**

controls data tips and drill down generation. Data tips are pieces of explanatory text that appear when you hold the mouse pointer over the data portions of a graph contained in an HTML page.

**IMAGEMAP**

specifies to generate data tips.

**NOIMAGEMAP**

specifies not to generate data tips.

**IMAGEMAP= ON | OFF**

controls data tips generation.

**ON**

specifies to generate data tips.

**OFF**

specifies not to generate data tips.

**Default**

NOIMAGEMAP or IMAGEMAP=OFF | NO

**Restrictions**

This option applies only when one of the ODS HTML* destinations is used.

Prior to **SAS 9.4M5**, an image map is not generated using SVG with ODS Graphics. The image map data that is used to produce tooltips and links is written directly in the SVG and is not part of the HTML. Using HTML5 with the inline SVG mode (the default value), the tooltips and links are written in the SVG portion of the document. (If you are using SAS 9.4M5, see the following Note.)

**Interaction**

When IMAGEMAP | IMAGEMAP=ON is specified and the ODS HTML destination is used, the IMAGE_DPI option in the ODS HTML destination is ignored, if specified, and the default image resolution of 96 DPI is used.

**Note**

Starting with **SAS 9.4M5**, image maps are supported with SVG output using HTML5. However, image maps are supported only when the HTML5 SVG mode is INLINE (the default value).

**IMAGENAME=”filename”**

specifies the base image filename. If more than one image is generated, each is assigned a filename that consists of a base name followed by a number in order to create unique names. This numbering can be reset with the RESET=INDEX option. Path information (if needed) can be set with the GPATH= option on the ODS destination statement. The default path is the current output directory. A file extension for filename is automatically generated based on the OUTPUTFMT= option.

**Note:** Starting with **SAS 9.4M5**, if a BY statement is in effect, you can uniquely name image files based on BY groups. To do this, insert #BY text into the image
name. For more information, see “Substituting BY Line Values in the Image Name” on page 1505.

| Default | The name of the output object. |
| Restriction | filename must be a single name. It must not include any path specification or image-format name extension. |
| Requirement | You must enclose filename in quotation marks. |
| See | “Specifying and Resetting the Image Name” on page 1505 |

**LABELMAX=**\(n\)
specifies the maximum number of labeled areas before data label collision avoidance is disabled. For example, if there are more than 50 points to be labeled and LABELMAX=50, then collision avoidance is turned off and the labels are all displayed at the top right of the data points.

\(n\)
specifies a positive integer.

| Default | 200 |
| Restriction | Data label collision avoidance is turned off under the following conditions:
- The number of observations with nonmissing labels exceeds the value specified by LABELMAX=.
- The number of observations exceeds five times the value specified by LABELMAX=.
A message is then sent to the SAS log. |
| Tip | To turn off collision avoidance, specify LABELMAX=0. |

**LABELPLACEMENT=** GREEDY | SA
specifies the label-placement algorithm to use for positioning labels in the graphs. The following labels are affected:
- data labels for needle plots, scatter plots, series plots, step plots, and vector plots
- vertex labels for line charts
- curve labels when the curve label is positioned at the start or end of the curve

**GREEDY**
specifies the Greedy method for managing label collision. The Greedy method tries different placement combinations in order to find an optimal approximation that avoids collisions. Label placement using this method is often less optimal than label placement using the Simulated Annealing (SA) method. However, depending on the number of data points and the potential for label collisions, the Greedy process can be significantly faster.

**SA**
specifies the Simulated Annealing method for managing label collision. The SA method attempts to determine the global minimization-of-cost function, which is based on a simulated annealing algorithm. The resulting label placement is usually better than placement using the Greedy method.
However, depending on the number of data points and the potential for label collisions, the SA method can be significantly slower.

**Restriction**
For BANDPLOT and LINECHART, the SA method has no effect on the curve labels when the CURVELABELPOSITION= option specifies START or END.

**Default**  
**LEGENDAREAMAX=**

specifies an integer that is interpreted as the maximum percentage of the overall graphics area that a legend can occupy.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LEGENDAREAMAX=</strong></td>
<td>(n)</td>
</tr>
</tbody>
</table>

\(n\) specifies a positive integer.

**Alias**  
**MAXLEGENDAREA=**

**Default**  
20

**Range**  
0–100

**Tip**  
To turn off the legend, specify **LEGENDAREAMAX=0**. No warning is issued when the legend is turned off in this way.

**LINEPATTERNOBSMAX=**

specifies the maximum the number of vertices for a patterned line. If the number of vertices exceeds the specified limit, the plot is not drawn.

\(n\) specifies a positive integer.

**Note:** This feature applies to SAS 9.4M5 and later releases.

**Default**  
10000

**Notes**  
If the number of vertices exceeds the specified limit, the plot is not drawn and a note is written to the SAS log. In that case, increase **LINEPATTERNOBSMAX=** to the value suggested in the log message, or change the line pattern to **SOLID**.

This option applies only to graphs created with the Graph Template Language and with the ODS Graphics procedures.

**LOESSOBSMAX=**

specifies an upper limit for the number of observations that can be used with a loess plot.
Note: Starting with SAS 9.4M3, LOESSOBSMAX= replaces LOESSMAXOBS=. However, LOESSMAXOBS= is supported as an alias. It is recommended that you use LOESSOBSMAX=.

If the number of observations of the loess plot exceeds the specified limit, the loess plot is not drawn.

For example, the following specifies that the most observations a loess plot can have is 1000.

LOESSOBSMAX=1000

Alias LOESSMAXOBS=

Default 5000

**NBINSMAX=** \( n \)

specifies the maximum number of bins that are processed for histograms. This option affects both computed and parameterized histograms, when available.

**Note:** This feature applies to SAS 9.4M4 and later releases.

\( n \) specifies a positive integer.

**Default** 10000

**Interaction** If you specify the number of bins in the histogram plot statement, that option is honored regardless of the NBINSMAX= option.

**NXYBINSMAX=** \( n \)

specifies the maximum number of bins that are processed for heat maps. This option affects both computed and parameterized heat maps. The option applies to the product of the X and Y bins.

**Note:** This feature applies to SAS 9.4M4 and later releases.

\( n \) specifies a positive integer.

**Default** 100000

**Interaction** If you specify the number of bins in the heat map statement, that option is honored regardless of the NXYBINSMAX= option.

**OBSMAX=** \( n \)

specifies the maximum number of observations that are processed. If the number of observations in the data set exceeds the value specified for OBSMAX=, the procedure step terminates with a log message.

\( n \) specifies a positive integer.

**Alias** MAXOBS=

**Default** 2 million observations
There are other ways to control the number of observations:

- **CASDATALIMIT=** system option,
- **DATALIMIT=** option in the CAS LIBNAME statement, and
- **DATALIMIT=** data set option.

If the CAS data transfer limit is set lower than OBSMAX=, then OBSMAX= has no effect.

### OUTPUTFMT=\textit{file–type} | STATIC

specifies the output format used to generate image or vector graphic files. If the image or vector graphic format is not valid for the active output destination, the format is automatically changed to the default format for that destination.

\textit{file–type} is the image or vector graphic format to be generated. See “Supported File Types for Output Destinations” on page 1510.

**STATIC** uses the best quality static image format for the active output destination. This is the default output format.

#### TIP
The STATIC keyword can be used to reset the output format to its default state.

**Default:** STATIC

**See:** “Specifying the Image Format” on page 1507

### PANELCELLMAX=\textit{n}

specifies the maximum number of cells in a graph panel where the number of cells is determined dynamically by classification variables. If the number of cells in the panel exceeds the specified limit, the panel is not drawn.

\textit{n} specifies a positive integer.

**Default:** 1000

**Note:** Graphs with DataPanel or DataLattice templates layouts are affected. In the ODS Graphics Procedures, this option affects graphs that are created with the SGPANEL procedure. If the value specified by the PANELCELLMAX= option is exceeded by any of these layouts, an empty graph is rendered and a warning message is issued.

**Tip:** You can use the TOTALCELLMAX= option to control the total maximum number of cells in the graph.

### PUSH | POP

pushes and pops ODS GRAPHICS settings in a stack. This feature enables you to temporarily save your custom settings in a stack and later restore those settings.

**Note:** This feature applies to SAS 9.4M3 and later releases.

**PUSH**

pushes the current ODS GRAPHICS settings to a stack.
POP
restores the most recently pushed settings from the stack. For each PUSH action, you can specify a POP request. ODS issues a warning if you specify POP without a corresponding PUSH. In that case, nothing is popped because nothing has been pushed.

The pushed settings remain in the stack in the current SAS session until they are popped or the stack is emptied.

Interaction You can specify PUSH as many times as you like up to the limit that is defined by the STACKDEPTHMAX= option. You can also use STACKDEPTHMAX= to empty the stack. For more information, see “Managing the Stack Depth” on page 1514.

Note Order of specification is important when using the PUSH and POP options. For more information, see “About PUSH and POP” on page 1512.

Tip Use the SHOW option to show the current ODS GRAPHICS settings.

See “Temporarily Saving and Restoring ODS GRAPHICS Settings” on page 1512

RESET | RESET= option
Reset one or more ODS GRAPHICS options to its default.

RESET
resets all options to their defaults.

RESET=
resets one of the following to its default:

ALL
resets all reset-options to their defaults.

ANTIALIAS
resets the ANTIALIAS= option to its default.

See ANTIALIAS= on page 1486

ANTIALIASMAX
resets the ANTIALIASMAX= option to its default.

See ANTIALIASMAX= on page 1486

ATTRPRIORITY
resets the ATTRPRIORITY= option to its default.

See ATTRPRIORITY= on page 1486

BORDER
resets the BORDER= option to its default.

See BORDER= on page 1487

BYLINE
resets the BYLINE= option to its default.

See BYLINE= on page 1487
DATASKINMAX
resets the DATASKINMAX= option to its default.

See DATASKINMAX= on page 1488

DISCRETEMAX
resets the DISCRETEMAX= option to its default.

See DISCRETEMAX= on page 1489

DRILLTARGET
resets the DRILLTARGET= option to its default.

See DRILLTARGET= on page 1489

GROUPMAX
resets the GROUPMAX= option to its default.

See GROUPMAX= on page 1490

HEIGHT
resets the HEIGHT= option to its default.

See HEIGHT= on page 1490

IMAGEMAP
resets the IMAGEMAP= option to its default.

Note Not all output destinations support this feature.

See IMAGEMAP= on page 1491

IMAGENAME
resets the IMAGENAME= option to its default.

Note: This feature applies to SAS 9.4M3 and later releases.

See IMAGENAME= on page 1491

INDEX <$\text{positive-integer}$>
resets the index counter that is appended to static image files.

When specifying this option, you can also specify the value for the index counter. The number that you specify must be enclosed in parentheses. $\text{positive-integer}$ determines the suffix for the next subsequent image, and increments with each new image. This feature applies to SAS 9.4M3 and later releases.

See “Resetting the Image Name” on page 1505

LABELMAX
resets the LABELMAX= option to its default.

See LABELMAX= on page 1492

LABELPLACEMENT
specifies the label-placement algorithm to use for positioning labels in the graphs.
LABELPLACEMENT resets the LABELPLACEMENT= option to its default.

See LABELPLACEMENT= on page 1492

LEGENDAREAMAX resets the LEGENDAREAMAX= option to its default.

See LEGENDAREAMAX= on page 1493

LOESSOBSMAX resets the LOESSOBSMAX= option to its default.

See LOESSOBSMAX= on page 1493

NBINSMAX resets the NBINSMAX= option to its default.

See NBINSMAX= on page 1494

NXYBINSMAX resets the NXYBINSMAX= option to its default.

See NXYBINSMAX= on page 1494

OUTPUTFMT resets the OUTPUTFMT= option to its default.

Note: This feature applies to SAS 9.4M3 and later releases.

See OUTPUTFMT= on page 1495

PANELCELLMAX resets the PANELCELLMAX= option to its default.

See PANELCELLMAX= on page 1495

SCALE resets the SCALE= option to its default.

See SCALE= on page 1499

SCALEMARKERS resets the SCALEMARKERS= option to its default.

See SCALEMARKERS= on page 1499

STACKDEPTHMAX resets the STACKDEPTHMAX= option to its default.

Note: This feature applies to SAS 9.4M3 and later releases.

See STACKDEPTHMAX= on page 1501

SUBPIXEL resets the SUBPIXEL option to its default.
Note: This feature applies to SAS 9.4M3 and later releases.

See SUBPIXEL on page 1502

TIPMAX
resets the TIPMAX= option to its default.

See TIPMAX = on page 1502

WIDTH
resets the WIDTH= option to its default.

See WIDTH= on page 1503

SCALE | NOSCALE | SCALE=ON | OFF
specifies whether the content of a graph is scaled proportionally.

SCALE
scales the components of a graph proportionally.

NOSCALE
does not scale the components of a graph proportionally.

SCALE=ON | OFF
specifies whether the content of a graph is scaled proportionally.

ON
scales the components of a graph proportionally.

Alias YES

OFF
does not scale the components of a graph proportionally.

Aliases NOSCALE

NO

Default SCALE or SCALE=ON | YES

SCALEMARKERS | NOSCALEMARKERS | SCALEMARKERS=ON | OFF
specifies whether the plot markers are to be scaled with the graph size. The scaling factor is based on the height of the graph cells and the height of the graph.

SCALEMARKERS
scales the markers with the graph size.

NOSCALE
does not scale the markers with the graph size.

SCALEMARKERS=ON | OFF
specifies whether the plot markers are to be scaled with the graph size.

ON
scales the markers with the graph size.

Alias YES
OFF
does not scale the markers with the graph size.

Aliases
NOSCALE
NO

Default
SCALEMARKERS or SCALEMARKERS=ON | YES

Restriction
Scaling is done only if the graph contains multiple cells or single nested cells.

SHOW
writes the current ODS Graphics settings to the SAS log. This option enables you to verify which settings are in effect. The option is especially useful when you use the PUSH and POP options to restore settings. For more information, see “Temporarily Saving and Restoring ODS GRAPHICS Settings” on page 1512.

Note: This feature applies to SAS 9.4M3 and later releases.

If no options have been specified, then SHOW lists those options for which ODS currently knows the default values.

The following statement resets all settings and shows the default values.

ods graphics / reset=all show;

Here are the default values displayed in the SAS log:

ODS Graphics Settings
---------------------
Output format: STATIC
By line: NOBYLINE
Antialias: ON
Maximum Loess observations: 5000
Maximum stack depth: 1024
Stack depth: 0
MaxObs: 2000000
Maximum Histogram Bins: 10000
Maximum Heatmap Bins: 100000
Maximum Obs for Patterned Lines: 10000
Maximum Total Cells per BY-group: 2000

If you have specified the settings for one or more options, then SHOW includes those settings along with the defaults.

Order of specification is important when using the SHOW option. For example, the following statement shows the current settings and then sets the NOBORDER option.

ods graphics / show noborder;

However, the following statement sets the NOBORDER option and then shows the settings. The NOBORDER setting is shown in the log along with the other settings that are in effect.

ods graphics / noborder show;
The following statement resets all settings. It then sets the image width and shows the default settings along with the specified width.

```ods graphics / reset=all width=5in show;```

Here are the default values plus the image width, as displayed in the SAS log:

```
ODS Graphics Settings
---------------------
Output format:          STATIC
By line:                NOBYLINE
Antialias:              ON
Maximum Loess observations: 5000
Image width:            5in
Maximum stack depth:    1024
Stack depth:            0
MaxObs:                 2000000
Maximum Histogram Bins: 10000
Maximum Heatmap Bins:   100000
Maximum Total Cells per BY-group: 2000
```

**Tip** If you have specified the settings for some options but want to see the default values without losing your specified settings, issue the following two statements. The first statement pushes your specified settings, resets all settings, and then lists options for which ODS currently knows the default values. The second statement restores your previous settings.

```ods graphics / push reset=all show;
ods graphics / pop;```

**STACKDEPTHMAX=n**
specifies the maximum stack depth for PUSH and POP requests. The stack is used to temporarily store ODS GRAPHICS settings when you issue PUSH requests. PUSH saves the current settings to the stack and increments the stack depth. POP restores the most recently saved settings from the stack and decrements the stack depth.

**Note:** This feature applies to SAS 9.4M3 and later releases.

`n` specifies a positive integer.

If `n` is less than the current stack depth, then the stack is popped until its depth equals `n`. Popping the stack does not affect other option settings.

**Defaults**

- 1024 is the default maximum depth
- 0 is the default depth

**Tips**

To empty the stack and then reset it to the default maximum depth, issue the following statement:

```ods graphics / stackdepthmax=0 reset=stackdepthmax;```

You can use any of the following commands to reset the stack to its default maximum depth:

```reset=stackdepthmax
reset=all
reset```
stackdepthmax=1024

See “Managing the Stack Depth” on page 1514

**SUBPIXEL | NOSUBPIXEL | SUBPIXEL=ON | OFF**
specifies whether subpixel rendering should be used for rendering ODS Graphics. Subpixel rendering produces smoother curves and more precise bar spacing.

*Note:* This feature applies to *SAS 9.4M3* and later releases.

**SUBPIXEL**
always uses subpixel rendering, when applicable, for rendering lines and bars.

**NOSUBPIXEL**
ever uses subpixel rendering.

**SUBPIXEL=ON | OFF**
specifies whether subpixel rendering should be used.

**ON**
always uses subpixel rendering, when applicable, for rendering lines and bars.

Alias YES

**OFF**
ever uses subpixel rendering.

Alias NO

**Default**
Subpixel rendering is always enabled for vector-graphics output. It is enabled by default for image output, unless the graph contains a scatter plot or a scatter-plot matrix. In those cases, subpixel rendering is disabled by default.

**Requirement**
Antialiasing must be enabled for this option to have any effect. Antialiasing is enabled by default. To re-enable antialiasing, use the ANTIALIAS=ON option in the ODS GRAPHICS statement.

**Tip**
For a large amount of data, antialiasing is disabled when the number of observations exceeds the default maximum of 4000 observations. In that case, subpixel rendering is also disabled. To increase the maximum, use the ANTIALIASMAX= option in the ODS GRAPHICS statement.

See “Subpixel Rendering” in *SAS ODS Graphics: Procedures Guide*

**TIPMAX=n**
specifies the maximum number of distinct data tip areas allowed before data tips are disabled. For example, if there are more than 400 points in a scatterplot, and TIPMAX=400, then no data tips appear. The default maximum value is 500.

*Note:* Prior to SAS 9.4M3, the TIPMAX= option specifies the maximum number of observations in the graph data to be allowed before data tips are disabled. The default is 500. When the graph data contains more than 500 observations,
data tips are disabled for the entire graph. Starting with SAS 9.4M3, the TIPMAX= option specifies the maximum number of data-tip areas allowed before data tips are disabled. This threshold is applied separately for each plot. The default remains at 500. If any plot in a graph contains more than 500 data-tip areas, data tips are disabled for that plot. Data tips are enabled for the remaining plots in the graph.

\[ n \]

specifies a positive integer.

Default 500

**TOTALCELLMAX=**\( n \)
specifies the maximum number of total cells in a graph where the number of cells is determined dynamically by classification variables.

Note: This feature applies to SAS 9.4M5 and later releases.

If the number of cells exceeds the specified limit, the graph is not drawn. This option is useful when multiple classification variables or large amounts of data result in a large number of cells.

\[ n \]

specifies a positive integer.

Default 2000

Notes

If the number of cells exceeds the limit specified for TOTALCELLMAX=, the graph is not drawn and an error message is written to the SAS log. The error message includes a suggested value for TOTALCELLMAX=. You can use the TOTALCELLMAX= option to increase the maximum number of cells. Note, however, that the processing time to render the graph increases with an increase in the maximum number of cells.

Graphs with DataPanel or DataLattice templates layouts are affected. In the ODS Graphics Procedures, this option affects graphs that are created with the SGPANEL procedure.

Tip

You can use the PANELCELLMAX= option to control the maximum number of cells in a graph panel.

**WIDTH=**\( \text{dimension} \)
specifies the width of any graph.

\( \text{dimension} \)
is a nonnegative number followed by one of these units of measure:

<table>
<thead>
<tr>
<th>Table A13.3</th>
<th>Units of Measure for Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>Centimeters</td>
</tr>
<tr>
<td>in</td>
<td>Inches</td>
</tr>
<tr>
<td>mm</td>
<td>Millimeters</td>
</tr>
<tr>
<td>pct or %</td>
<td>Percentage</td>
</tr>
</tbody>
</table>
Point size (72 points = 1 inch)

Defaults

The value of the SAS registry entry "ODS > ODS GRAPHICS > Design Width" or the value of the DesignWidth= option in a STATGRAPH template. Typically, this value is 640px.

For the PRINTER destination, units of 1/150 of an inch

Tips

If only the WIDTH= option is specified, then the default aspect of the graph is maintained.

When you change the size of a graph, by default graph elements such as markers, lines, text, and titles are scaled proportionally. This scaling takes effect even when you specify dimensions for the graph elements. To disable the scaling, specify the NOSCALE option.

Details

Using the ODS GRAPHICS Statement

You can enable ODS Graphics by using one of the following equivalent statements:

```sas
ods graphics on;
ods graphics;
```

When you specify one of these statements before your procedure invocation, Base, SAS/STAT, SAS/ETS, and SAS/QC procedures support ODS Graphics, either by default, or when you specify procedure options for requesting particular graphs.

To disable ODS Graphics, specify the following statement:

```sas
ods graphics off;
```

Note: ODS Graphics is ON by default for these procedures: SGDESIGN, SGMAP, SGPANEL, SGPIE, SGPLOT, SGRENDER, and SGSCATTER. For other products, the initial state of ODS Graphics is determined by a SAS Registry setting.

Using the ODS GRAPHICS Statement for Batch Jobs

To generate ODS Graphics output in UNIX batch jobs, you must set the DISPLAY system option before creating the output. To set the display, enter the following command:

```bash
export DISPLAY=<ip_address>:0
```

The `ip_address` is the TCP/IP address, or the name of a UNIX terminal. Usually, the IP address of the UNIX system where SAS is running would be used. If you do not set the DISPLAY variable, then you get an error message in the SAS log.
Specifying and Resetting the Image Name

Specifying the Image Name

For ODS Graphics output, by default, the ODS object name is used as the "root" name for the image output file. The following example creates a GIF image named REGPLOT:

```ods graphics / imagename="regplot" outputfmt=gif;
```

The assigned name REGPLOT is treated as a "root" name and the first output created is named REGPLOT. Subsequent graphs are named REGPLOT1, REGPLOT2, and so on, with an increasing index counter. This numbering can be reset with the RESET=INDEX option.

Resetting the Image Name

The RESET=INDEX option enables you to reset the filename numbering sequence. This feature applies to SAS 9.4M3 and later releases.

For a usage example, suppose that you are developing a paper or a presentation and it takes several submissions to get the desired output. You can use the RESET or RESET=INDEX option to force each output to replace itself:

```ods graphics / reset=index ... ;
```

This specification causes all subsequent images to be created with the default or current image name.

When specifying this option, you can also specify the value for the index counter. The value that you specify determines the suffix for the next subsequent image. For example:

```ods graphics / reset=index(100) imagename="MyName";
```

The next graph that you produce is named MYNAME100.

This feature is useful for creating animated graphics. For example, for a sequence of 100 images, you might begin with the following statement:

```ods graphics / reset=index(1) imagename="MyName";
```

In the example, your program produces 100 images named MYNAME1, MYNAME2, ..., MYNAME100. If you later add more images to the animation, you might submit the following:

```ods graphics / reset=index(101) imagename="MyName";
```

The next generated image is named MYNAME101.

Substituting BY Line Values in the Image Name

**Note:** This feature applies to SAS 9.4M5 and later releases.

If a BY statement is in effect for the data, then you can uniquely name image files based on BY groups. You do this by inserting any of the #BY* substitution items into the image filename. You can specify BY lines, BY variable values, or BY variable names for IMAGENAME=. You can combine the #BY* substitution item with other text. To do this, insert the #BY* item in the specified text string at the position where you want the substitution text to appear.

As explained in the IMAGENAME= description, the file extension is automatically generated based on the OUTPUTFMT= option.
Here are descriptions of the #BY* substitution items:

#BYLINE

substitutes the entire BY line without leading or trailing blanks for #BYLINE in the filename string. The BY line uses the format variable-name=value. However, the “=” character in the BY line is replaced with an underscore character.

Examples

The following example specifies the BY line as the image filename:

imagename="#byline"

The following example specifies the BY line preceded by the text sales_

imagename="sales_#byline"

#BYVALn | #BYVAL(BY-variable-name)

substitutes the current value of the specified BY variable for #BYVAL in the filename string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. You must provide the position even if your BY statement contains only one variable. Otherwise, no substitution occurs.

BY-variable-name
specifies a variable from the BY statement by its name. BY-variable-name is not case sensitive.

Examples

The following example specifies the first variable in the BY statement. The value of that variable is used for the image filename.

imagename="#byval1"

The following example specifies the second variable in the BY statement preceded by the text sales_

imagename="sales_#byval2"

The following example specifies the YEAR variable in the BY statement. The value of that variable is used for the image filename:

imagename="#byval(year)"

The following example specifies the YEAR variable in the BY statement followed with the text _sales:

imagename="#byval(year)_sales"

#BYVARn | #BYVAR(BY-variable-name)

substitutes the name of the BY variable or the label associated with the variable (whatever the BY line would normally display) for #BYVAR in the filename string. Specify the variable with one of these:

n
specifies a variable by its position in the BY statement. You must provide the position even if your BY statement contains only one variable. Otherwise, no substitution occurs.

BY-variable-name
specifies a variable from the BY statement by its name. BY-variable-name is not case sensitive.

Examples

The following example specifies the first variable in the BY statement. The name of that variable is used for the image filename.
The following example specifies the second variable in the BY statement preceded by the text `sales_`:
```plaintext
imagename="sales_#byvar2"
```

The following example specifies the YEAR variable in the BY statement. The name of that variable is used for the image filename:
```plaintext
imagename="#byvar(year)"
```

The following example specifies the YEAR variable in the BY statement followed with the text `_sales`:
```plaintext
imagename="#byvar(year)_sales"
```

**Note:** When using this feature, consider any filename length restrictions that might apply to your platform host. The filename length can vary greatly depending on the input data.

---

### Specifying the Image Format

Each ODS destination uses a default format for its output. You can use the `OUTPUTFMT=` option in the ODS GRAPHICS statement to change the output format.

**Note:** Unless you have a special requirement for changing the image format, we recommend that you not change it. The default PNG or vector graphic format is far superior to other formats, such as GIF, in support for transparency and a large number of colors. Also, PNG and vector graphics images require much less disk storage space than JPEG or TIFF formats.

If you want to generate vector graphics images, you can use the following `OUTPUTFMT=` values for each destination:

**Table A13.4 Generating Vector Graphics Output with ODS**

<table>
<thead>
<tr>
<th>ODS Destination</th>
<th><code>OUTPUTFMT=</code> Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS EPUB</td>
<td><code>OUTPUTFMT=SVG</code></td>
</tr>
<tr>
<td>ODS destination for Excel</td>
<td><code>OUTPUTFMT=EMF</code></td>
</tr>
<tr>
<td>ODS HTML</td>
<td><code>OUTPUTFMT=SVG</code></td>
</tr>
<tr>
<td>ODS LISTING</td>
<td><code>OUTPUTFMT=EMF</code></td>
</tr>
<tr>
<td></td>
<td><code>OUTPUTFMT=PDF</code></td>
</tr>
<tr>
<td></td>
<td>`OUTPUTFMT=PS</td>
</tr>
<tr>
<td></td>
<td><code>OUTPUTFMT=SVG</code></td>
</tr>
<tr>
<td></td>
<td><code>OUTPUTFMT=PCL</code></td>
</tr>
</tbody>
</table>
When a vector graphics image cannot be generated for the format that you specify, a PNG image is generated instead and is embedded in the specified output file. The output file format and extension are not changed in that case. In the following cases, a vector graphics image cannot be generated:

- surface plots
- bivariate histograms
- graphs that use smooth gradient contours
- graphs that include continuous legends
- graphs that use data skins
- graphs that use transparency (EMF and PS ODS destinations only)
- graphs that contain one or more rotated images
- graphs that have a broken axis
- graphs that contain outline marker characters

Starting with SAS 9.4M2, additional cases for which vector graphics output cannot be generated for graphs are as follows:

- graphs that use gradient fill for bars in a bar chart, histogram, or waterfall chart
- graphs that use the back-light effect on text
- graphs that include a text plot that displays text with an outlined bounding box or text with a filled bounding-box background
- graphs that include images (PostScript output only)

Starting with SAS 9.4M3, vector graphics output can be generated in the EMF, PDF, and SVG output formats for the following cases:

- graphs that use data skins
Note: For the EMF, PDF, and SVG formats, vector graphics output is not supported for graphs that use transparency and data skins. An image is generated in that case.

- graphs that include one or more rotated images
- graphs that use gradient fills (except PDF)
- graphs that use a continuous legend

Note: For the PDF output format, vector graphics output is not supported for graphs that use a continuous legend and data transparency. An image is generated in that case.

Starting with SAS 9.4M5, vector graphics output can or cannot be generated in the output formats as indicated in the following table:

**Table A13.5  Support for Vector Graphics in SAS 9.4M5**

<table>
<thead>
<tr>
<th>Case</th>
<th>EMF</th>
<th>PCL</th>
<th>PDF</th>
<th>PS</th>
<th>SVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface plots</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Gradient fill contour plots</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>3-D bivariate histograms</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Continuous legends without data or fill transparency</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Continuous legends with data transparency</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data skins</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data skins with transparency</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data transparency, without images</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Transparent images</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Images with data transparency</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Rotated images</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Text plots with backlight</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Text plots with fill and outline</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scatter plots with an outlined marker character</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Broken axis</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Fill patterns</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Supported File Types for Output Destinations

The following table lists all of the supported file types for some ODS output destinations.

Table A13.6  Supported File Types for Output Destinations

<table>
<thead>
<tr>
<th>Output Destination</th>
<th>Supported File Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPUB, EPUB2</td>
<td>PNG (default), GIF, JPG, SVG</td>
</tr>
<tr>
<td>EPUB3</td>
<td>SVG (default), PNG, GIF, JPG</td>
</tr>
<tr>
<td></td>
<td>Note: EPUB3 was added in SAS 9.4M1.</td>
</tr>
<tr>
<td></td>
<td>Note: Starting with SAS 9.4M3, EPUB3 is an alias for EPUB, and the EPUB3 supported</td>
</tr>
<tr>
<td></td>
<td>file types supersede the EPUB supported file types.</td>
</tr>
<tr>
<td>ODS destination for Excel</td>
<td>PNG (default), JPEG, JPG, EMF</td>
</tr>
<tr>
<td>HTML</td>
<td>PNG (default), GIF, JPEG, JPG, SVG</td>
</tr>
<tr>
<td>HTML5</td>
<td>SVG (default), PNG, GIF, JPEG, JPG</td>
</tr>
<tr>
<td>LISTING</td>
<td>PNG (default), BMP, EMF, EPSI, GIF, JFIF, JPEG, JPG, PDF, PS, SASEMF, STATIC, TIFF, WMF,</td>
</tr>
<tr>
<td></td>
<td>PSL, SVG</td>
</tr>
<tr>
<td>PDF</td>
<td>Native PDF (default), JPEG, JPG, GIF, PNG</td>
</tr>
<tr>
<td>ODS destination for PowerPoint</td>
<td>PNG (default), JPEG, JPG, GIF, EMF, TIFF, BMP</td>
</tr>
<tr>
<td>PS</td>
<td>PNG (default), JPEG, JPG, GIF, EPS, PDF, PCL, PS</td>
</tr>
<tr>
<td>RTF and Measured RTF</td>
<td>EMF (default), PNG, JPEG, JPG, JFIF</td>
</tr>
</tbody>
</table>

Description of Supported File Types

The following table provides descriptions of the supported file types for ODS output destinations.
<table>
<thead>
<tr>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMP (Microsoft Windows Device Independent Bitmap)</td>
<td>Supports color-mapped and true color images that are stored as uncompressed or run-length encoded data. BMP was developed by Microsoft Corporation.</td>
</tr>
<tr>
<td>CGM (Computer Graphics Metafile)</td>
<td>A free and open international standard file format for 2-D vector graphics, raster graphics, and text. This format is defined by ISO/IEC 8632.</td>
</tr>
<tr>
<td>DIB (Microsoft Windows Device Independent Bitmap)</td>
<td>See the description of BMP.</td>
</tr>
<tr>
<td>EMF Plus (Enhanced Metafile Format Plus Extensions)</td>
<td>Supports Enhanced Metafile Plus Extensions that provides additional functionality, such as support of RGBA colors.</td>
</tr>
<tr>
<td>EMF Dual (Enhanced Metafile Format and Enhanced Metafile Format Plus Extensions)</td>
<td>Produces both EMF and EMF Plus formats simultaneously in the same output.</td>
</tr>
<tr>
<td>EPS</td>
<td>Encapsulated PostScript</td>
</tr>
<tr>
<td>EPSI (Microsoft NT Enhanced Metafile)</td>
<td>An extended version of the standard PostScript (PS) format. Files that use this format can be printed on PostScript printers and can also be imported into other applications. Notice that EPSI files can be read, but PS files cannot be read.</td>
</tr>
<tr>
<td>GIF (Graphics Interchange Format)</td>
<td>Supports only color-mapped images. GIF is owned by CompuServe, Inc.</td>
</tr>
<tr>
<td>JFIF (JPEG File Interchange Format)</td>
<td>Supports JPEG image compression. JFIF software is developed by the Independent JPEG Group.</td>
</tr>
<tr>
<td>JPEG or JPG (Joint Photographic Experts Group)</td>
<td>A file format that is used for storing noninteractive images.</td>
</tr>
<tr>
<td>PBM (Portable Bitmap Utilities)</td>
<td>Supports gray-scale, color, RGB, and bitmap files. The Portable Bitmap Utilities are a set of free utility programs that were developed primarily by Jef Poskanzer.</td>
</tr>
<tr>
<td>File Type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PCL</td>
<td>Printer Control Language</td>
</tr>
<tr>
<td>PNG (Portable Network Graphic)</td>
<td>Supports true color, gray-scale, and 8-bit images.</td>
</tr>
<tr>
<td>PS (PostScript Image File Format)</td>
<td>The Image classes use only PostScript image operators. A level II PS printer is required for color images. PostScript was developed by Adobe Systems, Inc.</td>
</tr>
<tr>
<td>PSL (PostScript)</td>
<td>PostScript</td>
</tr>
<tr>
<td>STATIC</td>
<td>Chooses the best image format for the current ODS destination.</td>
</tr>
<tr>
<td>SVG (Scalable Vector Graphics)</td>
<td>Is an XML language for describing two-dimensional vector graphics.</td>
</tr>
<tr>
<td>TIFF (Tagged Image File Format)</td>
<td>Internally supports a number of compression types and image types, including bitmapped, color-mapped, gray-scaled, and true color. TIFF was developed by Aldus Corporation and Microsoft Corporation and is used by a wide variety of applications (available if licensed).</td>
</tr>
<tr>
<td>XBM</td>
<td>X Window Bitmap</td>
</tr>
<tr>
<td>XPM</td>
<td>X Window Pixmap</td>
</tr>
</tbody>
</table>

**Temporarily Saving and Restoring ODS GRAPHICS Settings**

**About PUSH and POP**

*Note: This feature applies to SAS 9.4M3 and later releases.*

Although you can use the RESET option to restore the default ODS GRAPHICS settings, there might be times when you want to save your current custom settings and later restore them. ODS enables you to temporarily store your custom settings in a stack created for this purpose, perform some other task with different settings, and then restore the previous settings.

The PUSH option saves the current ODS GRAPHICS settings to the stack and increments the stack depth. The POP option restores the most recently stored settings from the stack and decrements the stack depth.
This feature is useful when you write macros. Within a macro you can PUSH at the start of the macro and POP at the end. This enables your macro to have custom ODS GRAPHICS behaviors without affecting the calling environment.

You can specify PUSH as many times as you like up to the limit that is defined by the STACKDEPTHEMAX= option. The pushed settings remain in the stack in the current SAS session until they are popped or the stack is emptied. For more information, see “Managing the Stack Depth” on page 1514. For each PUSH option, you can specify a POP option. ODS issues a warning if you specify POP without a corresponding PUSH. In that case, nothing is popped because nothing has been pushed to the stack.

Order of specification is important when using the PUSH option. For example, the following statement pushes the NOBORDER option to the stack along with any other custom settings that are in effect.

```sas
ods graphics / noborder push;
```

A subsequent POP option restores the pushed settings including NOBORDER.

However, the following statement pushes the current custom settings and then sets the NOBORDER option.

```sas
ods graphics / push noborder;
```

Here, the subsequent POP option restores whatever border setting was in effect when the PUSH option was specified.

**TIP** Use the SHOW option to show the ODS GRAPHICS settings that are currently in effect.

Settings That Can Be Pushed

The PUSH and POP options apply to all ODS GRAPHICS options except the following: PUSH, POP, RESET=INDEX, and SHOW.

How Code Errors Affect the PUSH Operation

If the ODS GRAPHICS statement contains a syntax error, then the PUSH option is ignored.

For example, the PUSH option is ignored in the following statement:

```sas
ods graphics / antialias=bogus push;
```

A syntax error (BOGUS) in ANTIALIAS causes the parser to ignore the remaining options. However, a simple semantics error does not prevent the remaining options from being handled. In the following statement, the PUSH option is honored.

```sas
ods graphics / antialiasmax=-1 push;
```

In this statement, ANTIALIASMAX= –1 is invalid. The option expects a zero or a positive integer. In this case, a warning is issued to the log, but the PUSH occurs.

**Note:** Syntax errors in your code can have other unexpected results that are not described here.
Managing the Stack Depth

By default, the stack supports up to 1024 pushes. You can change the default by using the STACKDEPTHMAX= option.

If the specified STACKDEPTHMAX= value is less than the current stack depth, then the stack is popped until its depth equals the specified value. Popping the stack does not affect other option settings.

If you want to empty the stack, issue the following statement:

ods graphics / stackdepthmax=0 reset=stackdepthmax;

This statement first empties the stack of all PUSH requests and then restores the stack size to 1024.
Reserved Keywords and Unicode Values

Overview

The tables in this section show some of the reserved keywords and Unicode values that can be used with the UNICODE text command. For information about rendering Unicode characters, see "Managing the String on Text Statements" in *SAS Graph Template Language: User’s Guide*.

Note the following:

- Keywords and Unicode values are not case-sensitive: "03B1"x is the same code point as "03b1"x.
- The word *blank* is the keyword for a blank space.

### Lowercase Greek Letters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>α</td>
<td>03B1</td>
<td>lowercase alpha</td>
</tr>
<tr>
<td>beta</td>
<td>β</td>
<td>03B2</td>
<td>lowercase beta</td>
</tr>
<tr>
<td>gamma</td>
<td>γ</td>
<td>03B3</td>
<td>lowercase gamma</td>
</tr>
<tr>
<td>Keyword</td>
<td>Glyph</td>
<td>Unicode</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>---------</td>
<td>-----------------</td>
</tr>
<tr>
<td>delta</td>
<td>δ</td>
<td>03B4</td>
<td>lowercase delta</td>
</tr>
<tr>
<td>epsilon</td>
<td>ε</td>
<td>03B5</td>
<td>lowercase epsilon</td>
</tr>
<tr>
<td>zeta</td>
<td>ζ</td>
<td>03B6</td>
<td>lowercase zeta</td>
</tr>
<tr>
<td>eta</td>
<td>η</td>
<td>03B7</td>
<td>lowercase eta</td>
</tr>
<tr>
<td>theta</td>
<td>θ</td>
<td>03B8</td>
<td>lowercase theta</td>
</tr>
<tr>
<td>iota</td>
<td>ι</td>
<td>03B9</td>
<td>lowercase iota</td>
</tr>
<tr>
<td>kappa</td>
<td>κ</td>
<td>03BA</td>
<td>lowercase kappa</td>
</tr>
<tr>
<td>lambda</td>
<td>λ</td>
<td>03BB</td>
<td>lowercase lambda</td>
</tr>
<tr>
<td>mu</td>
<td>μ</td>
<td>03BC</td>
<td>lowercase mu</td>
</tr>
<tr>
<td>nu</td>
<td>ν</td>
<td>03BD</td>
<td>lowercase nu</td>
</tr>
<tr>
<td>xi</td>
<td>ξ</td>
<td>03BE</td>
<td>lowercase xi</td>
</tr>
<tr>
<td>omicron</td>
<td>ο</td>
<td>03BF</td>
<td>lowercase omicron</td>
</tr>
<tr>
<td>pi</td>
<td>π</td>
<td>03C0</td>
<td>lowercase pi</td>
</tr>
<tr>
<td>rho</td>
<td>ρ</td>
<td>03C1</td>
<td>lowercase rho</td>
</tr>
<tr>
<td>sigma</td>
<td>σ</td>
<td>03C3</td>
<td>lowercase sigma</td>
</tr>
<tr>
<td>tau</td>
<td>τ</td>
<td>03C4</td>
<td>lowercase tau</td>
</tr>
<tr>
<td>upsilon</td>
<td>υ</td>
<td>03C5</td>
<td>lowercase upsilon</td>
</tr>
<tr>
<td>phi</td>
<td>φ</td>
<td>03C6</td>
<td>lowercase phi</td>
</tr>
<tr>
<td>chi</td>
<td>χ</td>
<td>03C7</td>
<td>lowercase chi</td>
</tr>
<tr>
<td>psi</td>
<td>ψ</td>
<td>03C8</td>
<td>lowercase psi</td>
</tr>
<tr>
<td>omega</td>
<td>ω</td>
<td>03C9</td>
<td>lowercase omega</td>
</tr>
</tbody>
</table>
## Uppercase Greek Letters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha_u</td>
<td>Α</td>
<td>0391</td>
<td>uppercase alpha</td>
</tr>
<tr>
<td>beta_u</td>
<td>Β</td>
<td>0392</td>
<td>uppercase beta</td>
</tr>
<tr>
<td>gamma_u</td>
<td>Γ</td>
<td>0393</td>
<td>uppercase gamma</td>
</tr>
<tr>
<td>delta_u</td>
<td>Δ</td>
<td>0394</td>
<td>uppercase delta</td>
</tr>
<tr>
<td>epsilon_u</td>
<td>Ε</td>
<td>0395</td>
<td>uppercase epsilon</td>
</tr>
<tr>
<td>zeta_u</td>
<td>Ζ</td>
<td>0396</td>
<td>uppercase zeta</td>
</tr>
<tr>
<td>eta_u</td>
<td>Η</td>
<td>0397</td>
<td>uppercase eta</td>
</tr>
<tr>
<td>theta_u</td>
<td>Θ</td>
<td>0398</td>
<td>uppercase theta</td>
</tr>
<tr>
<td>iota_u</td>
<td>Ι</td>
<td>0399</td>
<td>uppercase iota</td>
</tr>
<tr>
<td>kappa_u</td>
<td>Κ</td>
<td>039A</td>
<td>uppercase kappa</td>
</tr>
<tr>
<td>lambda_u</td>
<td>Λ</td>
<td>039B</td>
<td>uppercase lambda</td>
</tr>
<tr>
<td>mu_u</td>
<td>М</td>
<td>039C</td>
<td>uppercase mu</td>
</tr>
<tr>
<td>nu_u</td>
<td>Ν</td>
<td>039D</td>
<td>uppercase nu</td>
</tr>
<tr>
<td>xi_u</td>
<td>Ξ</td>
<td>039E</td>
<td>uppercase xi</td>
</tr>
<tr>
<td>omicron_u</td>
<td>Ω</td>
<td>039F</td>
<td>uppercase omicron</td>
</tr>
<tr>
<td>pi_u</td>
<td>Π</td>
<td>03A0</td>
<td>uppercase pi</td>
</tr>
<tr>
<td>rho_u</td>
<td>Ρ</td>
<td>03A1</td>
<td>uppercase rho</td>
</tr>
<tr>
<td>sigma_u</td>
<td>Σ</td>
<td>03A3</td>
<td>uppercase sigma</td>
</tr>
<tr>
<td>tau_u</td>
<td>Τ</td>
<td>03A4</td>
<td>uppercase theta</td>
</tr>
<tr>
<td>upsilon_u</td>
<td>Υ</td>
<td>03A5</td>
<td>uppercase upsilon</td>
</tr>
<tr>
<td>phi_u</td>
<td>Φ</td>
<td>03A6</td>
<td>uppercase phi</td>
</tr>
<tr>
<td>chi_u</td>
<td>Χ</td>
<td>03A7</td>
<td>uppercase chi</td>
</tr>
</tbody>
</table>
### Reserved Keywords and Unicode Values

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>psi_u</td>
<td>Ψ</td>
<td>03A8</td>
<td>uppercase psi</td>
</tr>
<tr>
<td>omega_u</td>
<td>Ω</td>
<td>03A9</td>
<td>uppercase omega</td>
</tr>
</tbody>
</table>

### Special Characters

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Glyph</th>
<th>Unicode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prime</td>
<td>`</td>
<td>00B4</td>
<td>single prime sign</td>
</tr>
<tr>
<td>bar</td>
<td>−</td>
<td>0305</td>
<td>combining overline¹</td>
</tr>
<tr>
<td>bar2</td>
<td>=</td>
<td>033F</td>
<td>combining double overline¹</td>
</tr>
<tr>
<td>tilde</td>
<td>~</td>
<td>0303</td>
<td>combining tilde¹</td>
</tr>
<tr>
<td>hat</td>
<td>^</td>
<td>0302</td>
<td>combining circumflex accent¹</td>
</tr>
</tbody>
</table>

¹ This is an overstriking character that requires a Unicode font to render properly.
General Syntax for Attribute Options

Most statements provide options that enable you to specify attributes for the fills, lines, data markers, or text that is used in the display. For example, many plots provide a DATALABELATTRS= option that specifies the attributes of the data labels. This appendix discusses the general syntax for those options and the valid values for they accept.

A statement’s attribute options use the following general syntax:

```
attribute-option-name = style-element | style-element (options) | (options)
```

**style-element**

specifies the name of a style element. Only style attributes that are relevant for rendering the fill, line, data marker, or text are used.

**Example**

```
fillattrs = graphdata1
```
**style-element (options)**
specifies the name of a style element, plus individual options to be used as style overrides. Options is a space-delimited list of option = value pairs. Any options that are not specified are derived from the specified style element.

Example  
`fillattrs = graphdata1(color = blue)`

**(options)**
specifies individual options as a space-delimited list of option = value pairs. Any options that are not specified are derived from the default style element.

Example  
`datalabelattrs = (family = "Arial" size = 10pt)`

Depending on the attribute option used, the options might be fill options, line options, marker options, or text options.

In general, any relevant attribute that is not specified defaults to an internal value, which is typically derived from the style element that you specify for the attributes. When choosing a style element, you should use an element of the correct type. See “Graph Style Elements Used by ODS Graphics” in *SAS Graph Template Language: User’s Guide* for a list of style elements and their types.

---

**Attributes Available for the Attribute Options**

Depending on the attribute option used on a statement, the available attributes might be fill-color options on page 1520, fill-pattern options on page 1521, line options on page 1522, marker options on page 1523, or text options on page 1525.

---

**Fill Color Options**

Use one or more of the following options to specify fill-color attributes. The options must be enclosed in parentheses and specified as a space-delimited list of name = value pairs.

- **COLOR=style-reference | color**
  specifies the fill color. *Style-reference* must specify a valid style attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, or ENDcolor in the form style-element-name:attribute-name. If *style-reference* is not defined in the active ODS style, the COLOR= option is ignored and the default color is used.

  *Color* must be a valid color name, such as RED, or a color code, such as CXFF0000 or #FF0000. The color name must not exceed 64 characters. A color code must be a valid code for a SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB). See “Color-Naming Schemes” on page 1528.

- **TRANSPARENCY=number**
  specifies the degree of the transparency of the filled area. This setting enables you to set the transparency for the filled elements of some graph types. You can set just this fill transparency, or set the fill independently of the other transparent elements in the graph. For example, you can use this setting to set the
transparency level for the filled bars of a bar chart, and use the bar chart’s DATATRANSPARENCY= option to set a different transparency level for the bar outlines.

<table>
<thead>
<tr>
<th>Default</th>
<th>The same as the setting of the statement’s DATATRANSPARENCY= option.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Range</td>
<td>0–1, where 0 is opaque and 1 is entirely transparent</td>
</tr>
<tr>
<td>Interaction</td>
<td>This setting overrides the statement’s DATATRANSPARENCY= setting for the fills but not for the outlines.</td>
</tr>
<tr>
<td>Example</td>
<td>fillattrs = (transparency = 0.5)</td>
</tr>
</tbody>
</table>

## Fill Pattern Options

Use one or more of the following options to specify fill-pattern attributes. The options must be enclosed in parentheses and specified as a space-delimited list of name = value pairs.

**COLOR=** style-reference | color  
---|---  
specifies the fill color. *Style-reference* must specify a valid style attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, or ENDCOLOR in the form style-element-name:attribute-name. If *style-reference* is not defined in the active ODS style, the COLOR= option is ignored and the default color is used.

*Color* must be a valid color name, such as RED, or a color code, such as CXFF0000 or #FF0000. The color name must not exceed 64 characters. A color code must be a valid code for a SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB). See “Color-Naming Schemes” on page 1528.

**PATTERN=** style-reference | fill-pattern  
---|---  
specifies the fill pattern. *Style-reference* is valid starting with SAS 9.4M5. It must specify a FILLPATTERN attribute in the form style-element-name:FILLPATTERN. If *style-reference* is not defined in the active ODS style, the PATTERN= option is ignored and the default fill pattern is used, if one is specified by the active ODS style.

*Fill-pattern* is a two-character specification that consists of a line-direction prefix (R for right, L for left, and X for cross hatch) and a line-identification number, 1–5. The following table shows the patterns for each of the possible combinations.

### Table A15.1 Fill Patterns

<table>
<thead>
<tr>
<th>Pattern Name</th>
<th>Example</th>
<th>Pattern Name</th>
<th>Example</th>
<th>Pattern Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td><img src="image" alt="L1 Pattern" /></td>
<td>R1</td>
<td><img src="image" alt="R1 Pattern" /></td>
<td>X1</td>
<td><img src="image" alt="X1 Pattern" /></td>
</tr>
<tr>
<td>L2</td>
<td><img src="image" alt="L2 Pattern" /></td>
<td>R2</td>
<td><img src="image" alt="R2 Pattern" /></td>
<td>X2</td>
<td><img src="image" alt="X2 Pattern" /></td>
</tr>
</tbody>
</table>
Use one or more of the following options to specify line attributes. The options must be enclosed in parentheses and specified as a space-delimited list of name = value pairs.

**COLOR=style-reference | color**
specifies the line color. *Style-reference* must specify a valid style attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, or ENDCOLOR in the form style-element-name:attribute-name. If *style-reference* is not defined in the active ODS style, the COLOR= option is ignored and the default color is used.

*Color* must be a valid color name, such as RED, or a color code, such as CXFF0000 or #FF0000. The color name must not exceed 64 characters. A color code must be a valid code for a SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB). See “Color-Naming Schemes” on page 1528.

**PATTERN=style-reference | line-pattern-name | line-pattern-number**
specifies the line pattern. *Style-reference* must specify a LINESTYLE attribute in the form style-element-name:LINESTYLE. Line patterns can be specified as a pattern name or pattern number. The following table lists commonly used line patterns.

### Table A15.2 Commonly Used Line Patterns

<table>
<thead>
<tr>
<th>Pattern Number</th>
<th>Pattern Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Solid</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ShortDash</td>
<td>--------------</td>
</tr>
<tr>
<td>4</td>
<td>MediumDash</td>
<td>--------------</td>
</tr>
<tr>
<td>5</td>
<td>LongDash</td>
<td>--------------</td>
</tr>
<tr>
<td>8</td>
<td>MediumDashShortDash</td>
<td>--------------</td>
</tr>
</tbody>
</table>
Valid line-pattern numbers range from 1 to 46. Not all line-pattern numbers have names. See “Available Line Patterns” on page 1526 for a list of all possible line patterns. We recommend that you use the named patterns because they have been optimized to provide good discriminability when used in the same plot.

Note Anti-aliasing might alter the appearance of some line patterns that have fine detail such as line patterns 33 through 46. For example, if you specify the color black and the pattern 33 for a line, and anti-aliasing is enabled, then the line might appear gray. In that case, you can use the ODS GRAPHICS statement to disable anti-aliasing as shown in the following example:

```
ods graphics / antialias = off;
```

**THICKNESS=** *style-reference* | *dimension*

Specifies the line thickness. *Style-reference* must specify a LINETHICKNESS attribute in the form `style-element-name:LINETHICKNESS`.

See “*dimension*” on page 1480

**Marker Options**

Use one or more of the following options to specify data-marker attributes. You must enclose the options in parentheses and specify the options as a space-delimited list of `name = value` pairs.

**COLOR=** *style-reference* | *color*

Specifies the line color. *Style-reference* must specify a valid style attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, or ENDCOLOR in the form `style-element-name:attribute-name`. If *style-reference* is not defined in the active ODS style, the COLOR= option is ignored and the default color is used.
Color must be a valid color name, such as RED, or a color code, such as CXFF0000 or #FF0000. The color name must not exceed 64 characters. A color code must be a valid code for a SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB). See “Color-Naming Schemes” on page 1528.

Restriction This option has no effect on marker symbols that are created with the SYMBOLIMAGE statement.

See “SYMBOLIMAGE Statement” on page 1310

**SIZE=**<br>
Specifies the marker size (both width and height). Style-reference must specify a MARKERSIZE attribute in the form style-element-name:MARKERSIZE.

See “dimension” on page 1480

**SYMBOL=**<br>
Specifies the name of the marker. Style-reference must specify a MARKERSYMBOL attribute in the form style-element-name:MARKERSYMBOL. The following table lists the SAS symbols that are supported.

<table>
<thead>
<tr>
<th>Symbol Name</th>
<th>Plot Symbol</th>
<th>Symbol Name</th>
<th>Plot Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrowDown</td>
<td>↓</td>
<td>StarFilled</td>
<td>★</td>
</tr>
<tr>
<td>Asterisk</td>
<td>✶</td>
<td>Tack</td>
<td>T</td>
</tr>
<tr>
<td>Circle</td>
<td>◯</td>
<td>Tilde</td>
<td>∩</td>
</tr>
<tr>
<td>CircleFilled</td>
<td>●</td>
<td>Triangle</td>
<td>△</td>
</tr>
<tr>
<td>Diamond</td>
<td>◻</td>
<td>TriangleFilled</td>
<td>▲</td>
</tr>
<tr>
<td>DiamondFilled</td>
<td>◼</td>
<td>TriangleDown</td>
<td>▽</td>
</tr>
<tr>
<td>GreaterThan</td>
<td>＞</td>
<td>TriangleDownFilled</td>
<td>▼</td>
</tr>
<tr>
<td>Hash</td>
<td>#</td>
<td>TriangleLeft</td>
<td>△</td>
</tr>
<tr>
<td>HomeDown</td>
<td>▽</td>
<td>TriangleLeftFilled</td>
<td>◀</td>
</tr>
<tr>
<td>HomeDownFilled</td>
<td>◼</td>
<td>TriangleRight</td>
<td>△</td>
</tr>
<tr>
<td>IBeam</td>
<td>‹</td>
<td>TriangleRightFilled</td>
<td>◲</td>
</tr>
<tr>
<td>LessThan</td>
<td>＜</td>
<td>Union</td>
<td>△</td>
</tr>
</tbody>
</table>
You can also specify the names of symbols that are created by the SYMBOLCHAR and SYMBOLIMAGE statements.

See “SYMBOLCHAR Statement” on page 1303
“SYMBOLIMAGE Statement” on page 1310

**TRANSPARENCY=** *number*

specifies the degree of transparency for the plot markers.

Default: The transparency that is specified by the DATATRANSPARENCY= option, which is 0 by default.

Range: 0–1, where 0 is opaque and 1 is entirely transparent

Interaction: This suboption overrides the DATATRANSPARENCY= option for the plot markers only.

**WEIGHT=** *NORMAL | BOLD*

specifies the marker weight.

Restriction: This option has no effect on marker symbols that are created with the SYMBOLCHAR and SYMBOLIMAGE statements.

See “SYMBOLCHAR Statement” on page 1303
“SYMBOLIMAGE Statement” on page 1310

**Text Options**

Use one or more of the following options to specify text attributes. The options must be enclosed in parentheses and specified as a space-delimited list of *name* = *value* pairs.

**COLOR=** *style-reference | color*

specifies the line color. *Style-reference* must specify a valid style attribute such as COLOR, CONTRASTCOLOR, STARTCOLOR, NEUTRAL, or ENDCOLOR in the form *style-element-name:attribute-name*. If *style-reference* is not defined in the active ODS style, the COLOR= option is ignored and the default color is used.
Color must be a valid color name, such as RED, or a color code, such as CXFF0000 or #FF0000. The color name must not exceed 64 characters. A color code must be a valid code for a SAS color-naming scheme, such as RGB, CMYK, HLS, or HSV (HSB). See “Color-Naming Schemes” on page 1528.

**FAMILY=style-reference | "string"**
specifies the font family of the text. Style-reference must specify a FONTFAMILY attribute in the form `style-element-name:FONTFAMILY`.

**SIZE=style-reference | dimension**
specifies the font size of the text. Style-reference must specify a FONTSIZE attribute in the form `style-element-name:FONTSIZE`.

Restriction  The font size cannot be less than the minimum font size in SAS, which is determined by the SAS system. If you specify a font size that is less than the minimum font size, the minimum size is used.

See  “dimension” on page 1480

**STYLE=style-reference | NORMAL | ITALIC**
specifies the font style of the text. Style-reference must specify a FONTSTYLE attribute in the form `style-element-name:FONTSTYLE`.

**WEIGHT=style-reference | NORMAL | BOLD**
specifies the font weight of the text. Style-reference must specify a FONTWEIGHT attribute in the form `style-element-name:FONTWEIGHT`.

---

**Available Line Patterns**

The following line patterns can be used with the Graphics Template Language. A line pattern can be specified by its number or name. Not all patterns have names. We recommend that you use the named patterns because they have been optimized to provide good discriminability when used in the same plot.

**Table A15.4  Full List of Line Patterns**

<table>
<thead>
<tr>
<th>Pattern Number</th>
<th>Pattern Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Solid</td>
<td>----------------</td>
</tr>
<tr>
<td>2</td>
<td>ShortDash</td>
<td>----------------</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>----------------</td>
</tr>
<tr>
<td>4</td>
<td>MediumDash</td>
<td>----------------</td>
</tr>
<tr>
<td>5</td>
<td>LongDash</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern Number</td>
<td>Pattern Name</td>
<td>Example</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>MediumDashShortDash</td>
<td>- - - - - - - - - - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>DashDashDot</td>
<td>- - - - - - - - - - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>15</td>
<td>DashDotDot</td>
<td>- - - - - - - - - - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Dash</td>
<td>- - - - - - - - - - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>LongDashShortDash</td>
<td>- - - - - - - - - - - - - - - - - - - - - - -</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pattern Number</td>
<td>Pattern Name</td>
<td>Example</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Dot</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>ThinDot</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ShortDashDot</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>MediumDashDotDot</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Color-Naming Schemes

### Overview of Color-Naming Schemes

The valid color-naming schemes in SAS are as follows:
- RGB (red green blue)
- CMYK (cyan magenta yellow black)
- HLS (hue lightness saturation)
- HSV (hue saturation brightness), also called HSB
- Gray scale
- SAS color names (from the SAS Registry)
- SAS Color Naming System (CNS)

Table A4.5 on page 1529 shows examples of each color-naming scheme.

### Table A15.5 Examples of SAS Color-Naming Schemes

<table>
<thead>
<tr>
<th>Color-Naming Scheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>COLOR=(CX98FB98)</td>
</tr>
<tr>
<td>CMYK</td>
<td>COLOR=(FF00FF00)</td>
</tr>
<tr>
<td></td>
<td>COLOR=(CMYK00FFFF00)</td>
</tr>
<tr>
<td>HLS</td>
<td>COLOR=(H14055FF)</td>
</tr>
<tr>
<td>HSV</td>
<td>COLOR=(V0F055FF)</td>
</tr>
<tr>
<td>Gray Scale</td>
<td>COLOR=(GRAY4F)</td>
</tr>
<tr>
<td>SAS Registry Colors</td>
<td>COLOR=(palegreen)</td>
</tr>
<tr>
<td>CNS Color Names</td>
<td>COLOR=(VeryLightPurplishBlue)</td>
</tr>
<tr>
<td></td>
<td>COLOR=(Very_Light_Purplish_Blue)</td>
</tr>
</tbody>
</table>

You can also mix color-naming schemes in the same statement. For example:

```plaintext
DATACOLORS=(CXEE0044 vivid_blue darkgreen);
```

Each of the color-naming schemes has its advantages and disadvantages based on how the output is used. For example, if you are creating a report that will be viewed only online, then specifying colors using the RGB naming scheme or the SAS color names defined in the registry might produce better results. If you are creating a report for publishing in printed form, you might want to use the CMYK color-naming scheme.

**Note:** If you specify an invalid color name, the default color is used instead.

RGB Color Codes

The RGB color-naming scheme is usually used to define colors for a display screen. This color-naming scheme is based on the properties of light. An RGB color code defines a color by combining red, green, and blue colors in different ratios. All the colors combined together create white. The absence of all color creates black.

Color names are in the form CXrrggbb, where the following is true:

- CX indicates to SAS that this is an RGB color specification.
- rr is the red component.
- gg is the green component.
- bb is the blue component.

The components are hexadecimal numbers in the range 00–FF (0% to 100%). Each hexadecimal number indicates how much of the red, green, or blue is included in the color. Lower percentage values are darker and higher values are lighter. This scheme allows for up to 256 levels of each color component (more than 16 million different colors).

**Table A15.6  Examples of RGB Color Values**

<table>
<thead>
<tr>
<th>Color</th>
<th>RGB Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>CXFF0000</td>
</tr>
<tr>
<td>Green</td>
<td>CX00FF00</td>
</tr>
<tr>
<td>Blue</td>
<td>CX0000FF</td>
</tr>
<tr>
<td>White</td>
<td>CXXFFFFF</td>
</tr>
<tr>
<td>Black</td>
<td>CX000000</td>
</tr>
</tbody>
</table>

Any combination of the color components is valid. Some combinations match the colors produced by predefined SAS color names. See “Predefined Colors” in SAS Graph Template Language: User’s Guide.

CMYK Color Codes

The CMYK color-naming scheme is used in four-color printing. CMYK is based on the principles of objects reflecting light. Combining equal values of cyan, magenta, and yellow produces process black, which might not appear as pure black. The black component (K) of CMYK can be used to specify the level of blackness in the output. A lack of all colors produces white when the output is printed on white paper.

To specify the colors from a printer's Pantone Color Lookup Table, you can use the CMYK color-naming scheme. Specify colors in terms of their cyan, magenta, yellow,
and black components. Color names are in the form CMYKccmmyykk or Kccmmyykk, where the following is true:

- CMYK or K is an optional prefix that indicates to SAS that this is a CMYK color specification.
- cc is the cyan component.
- mm is the magenta component.
- yy is the yellow component.
- kk is the black component.

The color-value components are hexadecimal numbers in the range 00–FF, where higher values are darker and lower values are brighter. This scheme allows for up to 256 levels of each color component. When the color value begins with a letter (A–F), you can omit the CMYK or K prefix. When the color value begins with a number (0–9), you must use the CMYK or K prefix.

**Table A15.7 Examples of CMYK Color Values**

<table>
<thead>
<tr>
<th>Color</th>
<th>CMYK Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>CMYK00FFFF00</td>
</tr>
<tr>
<td>Green</td>
<td>FF00FF00</td>
</tr>
<tr>
<td>Blue</td>
<td>CMYKFFFF0000</td>
</tr>
<tr>
<td>White</td>
<td>K00000000</td>
</tr>
<tr>
<td>Process black (using cyan, magenta, and yellow ink)</td>
<td>FFFFFF00</td>
</tr>
<tr>
<td>Pure black (using only black ink)</td>
<td>K000000FF</td>
</tr>
</tbody>
</table>

**Note:** You can specify a CMY value by specifying zero (00) for kk, the color's black component.

CMYK color specifications are for output devices that support four colors. If you specify CMYK colors on an output device that supports three colors such as RGB, the CMYK colors are converted to colors in the three-color space. Because the four-color space supports many more colors than a three-color space, the CMYK colors might map to different colors in the three-color space. To preserve your CMYK colors in that case, specify a device that supports the CMYK color space. See “Color Support for Universal Printers” in *SAS Language Reference: Concepts*.

**HLS Color Codes**

The HLS color scheme specifies colors in terms of hue, lightness, and saturation levels. It is based on the Tektronix Color Standard. HLS color names are of the form Hhhlss, where the following is true:
H indicates to SAS that this is an HLS color specification.

$hhh$ is the hue component, which is expressed as an angle.

$l$ is the lightness component.

$s$ is the saturation component.

The components are hexadecimal numbers. The hue component is in the range 000–168 hexadecimal, which represents an angular value in the range 0–360 decimal. Hue starts with the primary color blue at 0 degrees, and then progresses to red at 120 degrees, to green at 240 degrees, and back to blue at 360 degrees. Both the lightness and saturation components are in the range 00–FF hexadecimal (0–255 decimal), which provides 256 levels that represent 0% to 100% for each component.

Table A15.8  Examples of HLS Color Codes

<table>
<thead>
<tr>
<th>Color</th>
<th>HLS Color Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>H07880FF</td>
</tr>
<tr>
<td>Green</td>
<td>H0F080FF</td>
</tr>
<tr>
<td>Blue</td>
<td>H00080FF</td>
</tr>
<tr>
<td>Light gray</td>
<td>H000BB00</td>
</tr>
<tr>
<td>White</td>
<td>HxxxFF00, such as H000FF00</td>
</tr>
<tr>
<td>Black</td>
<td>Hxxx0000, such as H0000000</td>
</tr>
</tbody>
</table>

1  When the saturation is set to 00, the color is a shade of gray that is determined by the lightness value. Therefore, white is defined as HxxxFF00 and black as Hxxx0000, where xxx can be any hue.

HSV (or HSB) Color Codes

The HSV color-naming scheme specifies colors in terms of hue, saturation, and value (or brightness) components. HSV color names are of the form V$hhh$s$vv$, where the following is true:

- $V$ indicates to SAS that this is an HSV color specification.
- $h$ is the hue component, which is expressed as an angle.
- $s$ is the saturation component.
- $v$ is the value or brightness component.

The components are hexadecimal numbers. The hue component is in the range 000–168 hexadecimal, which represents an angular value in the range 0–360 decimal. Hue starts with the primary color red at 0 degrees, and then progresses to green at 120 degrees, to blue at 240 degrees, and back to red at 360 degrees. Both the saturation and value (brightness) components are in the range 00–FF hexadecimal (0–255 decimal), which provides 256 levels for each component.
Table A15.9  Examples of HSV (or HSB) Color Codes

<table>
<thead>
<tr>
<th>Color</th>
<th>HSV Color Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>V000FFFF</td>
</tr>
<tr>
<td>Green</td>
<td>V078FFFF</td>
</tr>
<tr>
<td>Blue</td>
<td>V0F0FFFF</td>
</tr>
<tr>
<td>Light gray(^1)</td>
<td>Vxxx00BB such as V07900BB</td>
</tr>
<tr>
<td>White(^1)</td>
<td>Vxxx00FF such as V07900FF</td>
</tr>
<tr>
<td>Black(^1)</td>
<td>Vxxx00000 such as V0790000</td>
</tr>
</tbody>
</table>

\(^1\) When the saturation is set to 00, the color is a shade of gray. The value component determines the intensity of gray level. The xxx value can be any hue.

Gray-Scale Color Codes

Gray-scale colors are specified using the word GRAY and a lightness value in the form GRAY\(hh\). The value \(hh\) is the lightness of the gray and is a hexadecimal number in the range 00–FF, which provides 256 levels on the gray scale.

Note: GRAY, without a lightness value, is a SAS color name defined in the SAS registry. (See “SAS Color Names and RGB Values in the SAS Registry” on page 1535.) Its value is CX808080. Invalid color specifications are mapped to GRAY.

The following figure shows the gray-scale color for each hexadecimal value.
Color Naming System (CNS) Values

For CNS, you specify a color value by specifying lightness, saturation, and hue, in that order, using the terms shown in the following table.

Table A15.10  Color Naming System Values

<table>
<thead>
<tr>
<th>Lightness</th>
<th>Saturation</th>
<th>Hue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>Gray</td>
<td>Blue</td>
</tr>
<tr>
<td>Very Dark</td>
<td>Grayish</td>
<td>Purple</td>
</tr>
<tr>
<td>Dark</td>
<td>Moderate</td>
<td>Red</td>
</tr>
<tr>
<td>Medium</td>
<td>Strong</td>
<td>Orange/Brown</td>
</tr>
<tr>
<td>Light</td>
<td>Vivid</td>
<td>Yellow</td>
</tr>
<tr>
<td>Very Light</td>
<td></td>
<td>Green</td>
</tr>
<tr>
<td>White</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Follow these rules when you are determining the CNS color name:

- You should not use the lightness values Black and White with saturation or hue values.
If you do not specify default values, medium is the default lightness value and vivid is the default saturation value.

Gray is the only saturation value that can be used without a hue.

Unless the color that you want is black, white, or some form of gray, you must specify at least one hue.

You can use one or two hue values in the CNS color name. When you use two hue values, the hues must be adjacent to each other in the following list: blue, purple, red, orange/brown, yellow, green, and then returning to blue. Two hue values result in a color that is a combination of both colors. Use the suffix "ish" to reduce the effect of a hue when two hues are combined. Reddish purple is less red than red purple. The color specified with the "ish" suffix must precede the color without the "ish" suffix.

You can write color names in the following ways:
- without space separators between words
- with an underscore to separate words

Do not enclose color names in quotation marks in GTL. For example, the following color specifications are valid:
- verylightmoderatepurplishblue
- very_light_moderate_purplish_blue

Note: If a CNS color name is also a color name in the SAS Registry, the SAS Registry color value takes precedence. Some CNS color names and color names in the SAS Registry have different color values. Dark blue is an example. To use a CNS color value when the color name is also in the SAS Registry, include an underscore to separate the words. Here is an example.

```
color=dark_blue
```

---

**SAS Color Names and RGB Values in the SAS Registry**

The SAS Registry provides a set of color names and RGB values that you can use to specify colors. These color names and RGB values are common to most web browsers. You can specify the name itself or the RGB value associated with that color name. To view the color names as associated RGB values that are defined in the registry, submit the following code:

```
proc registry list
  startat="COLORNAMES";
run;
```

SAS prints the output in the SAS log. Here is a partial listing.

```
NOTE: Contents of SASHELP REGISTRY starting at subkey [COLORNAMES]
[   COLORNAMES]
  Active="HTML"
[   HTML]
    AliceBlue=hex: F0,F8,FF
    AntiqueWhite=hex: FA,EB,D7
    Aqua=hex: 00,FF,FF
```

Aquamarine=hex: 7F,FD,D4
Azure=hex: F0,FF,FF
Beige=hex: F5,F5,DC
...

For a list of the color names that are defined in the SAS Registry, see “Predefined Colors” in SAS Graph Template Language: User’s Guide.

You can also create your own color values by adding them to the SAS registry. For information about viewing and modifying the list of color names, see SAS Language Reference: Concepts.

Converting Color Values between Color-Naming Schemes

If the SAS/GRAPH software is included in your SAS installation, several macros are provided with SAS/GRAPH that enable you to perform selected conversions between color-naming schemes. The following table shows the conversions that are possible using these macros.

<table>
<thead>
<tr>
<th>From Color Value</th>
<th>To Color Value</th>
<th>Conversion Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMY</td>
<td>RGB</td>
<td>%CMY</td>
</tr>
<tr>
<td>CNS</td>
<td>HLS</td>
<td>%CNS</td>
</tr>
<tr>
<td>RGB</td>
<td>HLS</td>
<td>%RGB2HLS</td>
</tr>
<tr>
<td>HLS</td>
<td>RGB</td>
<td>%HLS2RGB</td>
</tr>
</tbody>
</table>

For information about these macros, see SAS/GRAPH: Reference.

Converting Numeric Color Component Values to Color Names

If the SAS/GRAPH software is included in your SAS installation, several macros are provided with SAS/GRAPH that enable you to convert numeric color component values to color names. The following table shows the macros that you can use for this purpose.

<table>
<thead>
<tr>
<th>From Numeric Color Component Values</th>
<th>To Color Name</th>
<th>Conversion Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>cyan, magenta, yellow, black</td>
<td>CMYK</td>
<td>%CMY</td>
</tr>
<tr>
<td>hue, lightness, saturation</td>
<td>HLS</td>
<td>%HLS</td>
</tr>
<tr>
<td>hue, saturation, value</td>
<td>HSV</td>
<td>%HSV</td>
</tr>
</tbody>
</table>
## Converting Color Values from Other Applications

Many software programs enable you to change or customize various colors. A dialog box typically provides a means of selecting a different color or modifying the attributes of an existing color. In the SAS ODS Graphics Editor, for example, the More Colors dialog box shown in the following figure serves this purpose.

*Figure A15.1 More Colors Dialog Box*

Figure A4.1 on page 1537 shows the HSB and RGB numeric color component values for the currently selected color. It also shows the RGB values in hexadecimal. To use the RGB specification for this color in a SAS program, add the CX prefix to the Hex value E7B3B4. If the application provides only the numeric component values, you must convert the decimal component values to hexadecimal. In Figure A4.1 on page 1537, the H value is in the range 0–360 degrees, and the S and B values are each in the range 0–100 percent. The R, G, and B values are in the range 0–255 each. You can convert the component values manually. (See “Understanding Hexadecimal Values” in SAS Graph Template Language: User’s Guide.) Be aware that you must first convert the S and B values from percentages to 255-based values by rounding the result of the following computation to the nearest integer:

\[
\left( \frac{\text{Value}}{100} \right) \times 255
\]

If the SAS/GRAPH software is included in your SAS installation, you can also use the SAS/GRAPH color utility macros to convert the values. The %HSV color utility

<table>
<thead>
<tr>
<th>From Numeric Color Component Values</th>
<th>To Color Name</th>
<th>Conversion Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>red, green, blue</td>
<td>RGB</td>
<td>%RGB</td>
</tr>
</tbody>
</table>
macro converts HSV (HSB) numeric color component values to an HSV color name. Likewise, the %RGB macro converts RGB numeric color component values to an RGB color name. The following example shows how to use the %HSV color utility macro to convert the H, S, and B color component values in Figure A4.1 on page 1537 to an HSV color name.

```sas
%COLORMAC;
  data _null_
    put "%HSV(357,22,90)";
run;
```

Note: The %COLORMAC macro compiles all of the SAS/GRAPH color utility macros. You need to run it only once during a SAS session.

Because the %HSV macro accepts values in the range 0–100 as a percentage of 255 for saturation and value, use the S and B values as shown. The result is V16538E6.

The following example shows how to use the %RGB color utility macro to convert the R, G, and B numeric color component values in Figure A4.1 on page 1537 to an RGB color name.

```sas
/* Compute the RGB percentages */
data _null_
  r = 231;
  g = 179;
  b = 180;
  call symputx("r", round((r/255)*100));
  call symputx("g", round((g/255)*100));
  call symputx("b", round((b/255)*100));
run;

/* Convert to RGB color name */
%COLORMAC;
  data _null_
    put "%RGB(&r,&g,&b)";
run;
```

Because the %RGB color utility macro accepts integer values in the range 0–100 as a percentage of 255, you must convert the 255-based values to integer percentages in order to use them in the %RGB macro call. The result is CXE8B3B5. The result from the %RGB macro is not exact due to rounding. If you want more exact results, use the following program.

```sas
data _null_
  r = 231;
  g = 179;
  b = 180;
  rgb="CX" || put(r,hex2.) || put(g,hex2.) || put(b,hex2.);
  put rgb;
run;
```

The result is CXE7B3B4.

To convert the RGB color name to an HLS color name, use the %RGB2HLS macro as shown in the following program.

```sas
%COLORMAC;
  data _null_
    put "%RGB2HLS(CXE7B3B4)";
```
run;
The result is H077CD84.

For more information about the SAS/GRAPH color utility macros, see SAS/GRAPH: Reference.
Using SAS Formats

SAS formats can be assigned to input data columns with the FORMAT statement of the SGRENDER procedure. Also, several GTL statement options enable a SAS format as an option value. Examples include the TICKVALUEFORMAT= option for formatting axis tick values, and the TIPFORMAT= option for formatting data tips.

Not all SAS formats are supported in GTL or with the SGPLOT, SGSCATTER, SGPANEL, and SGRENDER procedures. The tables in the following sections show the character and numeric SAS formats that are not supported.

When GTL encounters an unsupported format, a note similar to the following is written to the SAS log:

TICKVALUEFORMAT=bestx. is invalid. The format is invalid or unsupported. The default will be used.

Unsupported Numeric Formats

The following numeric formats are not supported in GTL:

<table>
<thead>
<tr>
<th>Format</th>
<th>Format</th>
<th>Format</th>
<th>Format</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>BESTD</td>
<td>BESTX</td>
<td>D</td>
<td>FLOAT</td>
<td>FRACT</td>
</tr>
</tbody>
</table>
### Unsupported Date and Time Formats Related to ISO 8601

The following date and time formats are not supported in GTL:

<table>
<thead>
<tr>
<th>Format</th>
<th>Format</th>
<th>Format</th>
<th>Format</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N8601B</td>
<td>$N8601BA</td>
<td>$N8601E</td>
<td>$N8601EA</td>
<td>$N8601EH</td>
</tr>
<tr>
<td>$N8601EX</td>
<td>$N8601H</td>
<td>$N8601X</td>
<td>B8601DA</td>
<td>B8601DN</td>
</tr>
<tr>
<td>B8601DT</td>
<td>B8601DZ</td>
<td>B8601LZ</td>
<td>B8601TM</td>
<td>B8601TZ</td>
</tr>
<tr>
<td>E8601DA</td>
<td>E8601DN</td>
<td>E8601DT</td>
<td>E8601DZ</td>
<td>E8601LZ</td>
</tr>
<tr>
<td>E8601TM</td>
<td>E8601TZ</td>
<td>IS8601DA</td>
<td>IS8601DN</td>
<td>IS8601DT</td>
</tr>
<tr>
<td>IS8601DZ</td>
<td>IS8601LZ</td>
<td>IS8601TM</td>
<td>IS8601TZ</td>
<td></td>
</tr>
</tbody>
</table>

### Other Unsupported Date and Time Formats

The following date and time formats are not supported in GTL:

<table>
<thead>
<tr>
<th>Format</th>
<th>Format</th>
<th>Format</th>
<th>Format</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDATE</td>
<td>HEBDATE</td>
<td>JDATEMDW</td>
<td>JDATEMNW</td>
<td>JDATEWK</td>
</tr>
<tr>
<td>JDATEYDW</td>
<td>JDATEYM</td>
<td>JDATEYMD</td>
<td>JDATEYMW</td>
<td>JDATEYT</td>
</tr>
<tr>
<td>JDATEYTW</td>
<td>JNENGO</td>
<td>JNENGOT</td>
<td>JNENGOTW</td>
<td>JNENGOW</td>
</tr>
<tr>
<td>JTIMEH</td>
<td>JTIMEHM</td>
<td>JTIMEHMS</td>
<td>JTIMEHW</td>
<td>JTIMEMW</td>
</tr>
<tr>
<td>JTIMESW</td>
<td>MDYAMP</td>
<td>MINGUO</td>
<td>NENGO</td>
<td>NLDATEYQ</td>
</tr>
<tr>
<td>NLDATEYR</td>
<td>NLDATEYW</td>
<td>NLDATMYQ</td>
<td>NLDATMYR</td>
<td>NLDATMYW</td>
</tr>
<tr>
<td>NLSTRMON</td>
<td>NLSTRQTR</td>
<td>NLSTRWK</td>
<td>PDJULG</td>
<td>PDJULI</td>
</tr>
</tbody>
</table>
Unsupported Currency Formats

The following currency formats are not supported in GTL:

<table>
<thead>
<tr>
<th>EURFRATS</th>
<th>EURFRBEL</th>
<th>EURFRCHF</th>
<th>EURFRCZK</th>
<th>EURFRDEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>EURFRDKK</td>
<td>EURFRESP</td>
<td>EURFRFIM</td>
<td>EURFRFRF</td>
<td>EURFRGBP</td>
</tr>
<tr>
<td>EURFRGRD</td>
<td>EURFRHUF</td>
<td>EURFRIEP</td>
<td>EURFRTL</td>
<td>EURFRLUF</td>
</tr>
<tr>
<td>EURFRNLG</td>
<td>EURFRNOK</td>
<td>EURFRPLZ</td>
<td>EURFRPTE</td>
<td>EURFRROL</td>
</tr>
<tr>
<td>EURFRRUR</td>
<td>EURFRSEK</td>
<td>EURFRSIT</td>
<td>EURFRTRL</td>
<td>EURFRYUD</td>
</tr>
<tr>
<td>EURTOATS</td>
<td>EURTOBEF</td>
<td>EURTOCHF</td>
<td>EURTOCZK</td>
<td>EURTODEM</td>
</tr>
<tr>
<td>EURTODKK</td>
<td>EURTOESP</td>
<td>EURTOFIM</td>
<td>EURTOFRF</td>
<td>EURTOGBP</td>
</tr>
<tr>
<td>EURTOGRD</td>
<td>EURTOHUF</td>
<td>EURTOIEP</td>
<td>EURTOITL</td>
<td>EURTOLUF</td>
</tr>
<tr>
<td>EURTONLG</td>
<td>EURTONOK</td>
<td>EURTOPLZ</td>
<td>EURTOPTE</td>
<td>EURTOROL</td>
</tr>
<tr>
<td>EURTORUR</td>
<td>EURTOSEK</td>
<td>EURTOSIT</td>
<td>EURTOTRL</td>
<td>EURTOYUD</td>
</tr>
</tbody>
</table>

Unsupported User-Defined Formats

Picture Formats with Date, Time, or Datetime Directives

ODS Graphics does not support user-defined picture formats that specify date, time, or datetime directives such as %a, %W, %H, and so on. Using a picture format with directives produces unexpected results in ODS Graphics. To work around this limitation, apply the picture format to a new column in your data, and then plot the new preformatted column instead. Here is an example that creates column NewDate by formatting the Date column as MM/YYYY.

```/* Create picture format MonYear for MM/YYYY date */
proc format;
   picture monyear (min=7)
      low-high='%0m/%Y' (datatype=date);
run;```
/* Add preformatted date column NewDate to the data */
data stocks;
  set sashelp.stocks
    (where=(year(date) eq 2001 and stock eq "IBM"));
  newdate = put(date, monyear.);
  label newdate="Date: MM/YYYY";
run;

The NewDate column can be used in ODS Graphics plot statements instead of the Date column.

For information about the PICTURE statement in the FORMAT procedure, see Base SAS Procedures Guide.

Unicode Values

In SAS 9.4M2 and in earlier releases, ODS Graphics does not support Unicode values in user-defined formats. Starting with SAS 9.4M3, ODS Graphics supports Unicode values in user-defined formats only if they are preceded by the (*ESC*) escape sequence as shown in the following example.

"(*ESC*){unicode beta}"

ODS Graphics does not support the use of a user-defined ODS escape character to escape Unicode values in user-defined formats.

For an example of how to use Unicode values in user-defined formats with ODS Graphics, see “Formatting the Tick Values on a Discrete Axis” in SAS Graph Template Language: User’s Guide.
Generalized Macro for BOXPLOTPARM Data

The following SAS code is a generalized macro for computing input data for BOXPLOTPARM.

```sas
%macro boxcompute(indsn=,x=,y=,outdsn=boxdata,datalabel=, qntldef=5,table=no);
/* NOTE: INDSN, X and Y are required parameters, where
INDSN = input SAS data set
X = categorical variable (num or char)
Y = response variable (num)
OUTDSN = output dataset. It contains these variables:
   STAT: Statistic names for BOXPLOTPARM
   VALUE: values for STAT type
   X: X variable values
   DATALABEL: outlier labels from the DATALABEL= variable
   N, Mean, Median, Std if TABLE=YES
   DATALABEL= variable used to label outliers (num or char)
   QNTLDEF = 1|2|3|4|5
   (how to compute quantiles - see PROC SUMMARY)
   TABLE = YES | NO
   (add additional data to build table of statistics)
*/
%macro varinfo(dsid,varname,role,rc);
/* utility macro for obtaining variable info */
%local varnum;
%if %length(&varname)=0 %then %do;
%let &rc=0;  %return;
%end;
%let varnum=%sysfunc(varnum(&dsid,&varname));
%if &varnum > 0 %then %do;
   %let &role.label=%sysfunc(varlabel(&dsid,&varnum));
   %if %length(&&role.label)=0 %then
      %let &role.label=%sysfunc(varname(&dsid,&varnum));
   %let &role.fmt=%sysfunc(varfmt(&dsid,&varnum));
   %let &rc=0;
```

1545
%end;
%else %do;
  %put ERROR: %upcase(&role) variable &varname not found.;
  %let &rc=1;
%end;
%mend varinfo;

/* validate dataset and variables */
%local dsid ylabel xlabel datalabellabel yfmt xfmt datalabelfmt rc_y rc_x rc_d;
%let dsid=%sysfunc(open(&indsn));
%if &dsid %then %do;
  %varinfo(&dsid,&y,Y,rc_y)
  %varinfo(&dsid,&x,X,rc_x)
  %if %length(&datalabel) %then 
    %varinfo(&dsid,&datalabel,DATALABEL,rc_d);
  %else %let rc_d=0;
  %let dsid=%sysfunc(close(&dsid));
  %if &rc_y or &rc_x or &rc_d %then %return;
%end;
%else %do;
  %put ERROR: Input dataset &indsn not found.;
  %return;
%end;

/* compute basic summary statistics */
proc summary data=&indsn(rename=(&y=VALUE &x=X))
  nway qntldef=&qntldef;
  class x;
  var value;
  output out=summary(drop=_type_ _freq_) n=N mean=Mean
    median=Median q1=Q1 q3=Q3 std=STD / noinherit;
run;
proc sort data=&indsn(keep=&x &y &datalabel)
  %if %length(&datalabel) %then
    out=sorted(rename=(&x=X &y=VALUE &datalabel=DATALABEL))
  %else out=sorted(rename=(&x=X &y=VALUE));
  ; by &x;
run;
/* compute fences, MIN, MAX and any outliers for X values */
data outliers;
  length STAT $10;
  %if %length(&datalabel) %then
    %do;
    keep STAT X VALUE DATALABEL;
    label VALUE="&ylabel" X="&xlabel"
      DATALABEL="&datalabellabel";
    format VALUE &yfmt X &xfmt DATALABEL &datalabelfmt;
    %end;
  %else
    %do;
    keep STAT X VALUE;
    label VALUE="&ylabel" X="&xlabel";
    format VALUE &yfmt X &xfmt;
    %end;
  retain lowerFence upperFence farLowerFence farUpperFence
tempmin tempmax;
merge sorted summary; by x;

/* perform computations for each X value */
if first.x then do;
    lowerFence=q1-((q3-q1)*1.5);
    upperFence=q3+((q3-q1)*1.5);
    farLowerFence=q1-((q3-q1)*3);
    farUpperFence=q3+((q3-q1)*3);
    /* these computations for MIN and MAX result
    in the same values produced by the BOXPLOT
    statement, however they can be modified to
    satisfy other statistical definitions */
    if value <= upperFence then tempmax=value;
    else tempmax=.;
    if value >= lowerFence then tempmin=value;
    else tempmin=.;
end;

/* recompute MAX and MIN for each obs */
if 0 <= sum(upperFence,-value) then
    tempmax=max(tempmax,value);
if 0 <= sum(value,-lowerFence) then
    tempmin=min(tempmin,value);

/* write out both types of outliers */
if value < farLowerFence or value > farUpperFence then do;
    stat="FAROUTLIER"; output;
end;
else if value < LowerFence or value > UpperFence then do;
    stat="OUTLIER"; output;
end;

/* write out MIN and MAX for each X value */
if last.x then do;
    value=tempmin; stat="MIN"; output;
    value=tempmax; stat="MAX"; output;
end;
run;

/* transpose the stats into the columns
   required by BOXPLOTPARM */
data transpose(keep=x stat value);
    length STAT $10;
    set summary;
    array stats{*} n--std;
    do i=1 to dim(stats);
        stat=upcase(vname(stats{i}));
        VALUE=stats{i};
        output;
    end;
run;
/* interleave the obs by the X variable */
data &outdsn;
    set transpose outliers; by X;
run;
/* merge the output stats for building a stat table */
%if %upcase(&table)=YES %then %do;
    data &outdsn;
        merge &outdsn summary; by X;
%end;
Here is the macro invocation to produce the data for the graph shown in the section “Example: BOXPLOTPARM Statement” on page 411.

```sas
%boxcompute(indsn=sashelp.cars,x=type,y=mpg_city, datalabel=make);
```

```sas
proc template;
  define statgraph boxplotparm1;
  begingraph;
    entrytitle "City Mileage for Vehicle Types";
    layout overlay;
      boxplotparm y=value x=x stat=stat / datalabel=datalabel spread=true;
    endlayout;
  endgraph;
end;
run;
```

```sas
proc sgrender data=boxdata template=boxplotparm1;
run;
```

The following figure and code show an example of a “table” of statistics with BLOCKPLOT statements.

```sas
%boxcompute(indsn=sashelp.cars,x=type,y=mpg_highway, outdsn=boxdata2,table=yes);
```

```sas
proc template;
  define statgraph boxplotparm2;
  begingraph;
    ...
  endgraph;
end;
```

```
%boxcompute(indsn=sashelp.cars,x=type,y=mpg_highway, outdsn=boxdata2,table=yes);
```

```
proc template;
  define statgraph boxplotparm2;
  begingraph;
    ...
  endgraph;
end;
```
entrytitle "Highway Mileage for Vehicle Types";
layout overlay / xaxisOpts=(offsetMin=0.08 offsetMax=0.08);
innerMargin / align=top;
blockplot x=x block=n /
display=(values label outline) valuehalign=center
labelattrs=graphdatatext valueattrs=graphdatatext;
blockplot x=x block=std /
display=(values label outline) valuehalign=center
labelattrs=graphdatatext valueattrs=graphdatatext;
blockplot x=x block=mean /
display=(values label outline) valuehalign=center
labelattrs=graphdatatext valueattrs=graphdatatext;
blockplot x=x block=median /
display=(values label outline) valuehalign=center
labelattrs=graphdatatext valueattrs=graphdatatext;
endInnerMargin;
boxplotparm y=value x=x stat=stat /
datalabel=datalabel spread=true;
endlayout;
endgraph;
end;
run;

proc sgrender data=boxdata2 template=boxplotparm2;
run;
Appendix 7

Memory Management for ODS Graphics

SAS Options Affecting Memory ......................................................... 1551
Managing a Java Out of Memory Error ........................................... 1552

SAS Options Affecting Memory

ODS Graphics uses Java technology to produce its graphs. Most of the time this fact is transparent to you because the required Java Runtime Environment (JRE) and JAR files are included with SAS software installation. Also, the Java environment is automatically started and stopped for you. When Java is started, it allocates a fixed amount of memory. The memory can grow up to the value set for the -Xmx suboption in the JREOPTIONS option (discussed in a moment). This memory is independent of the memory limit that SAS sets for the SAS session with its MEMSIZE= option.

Normally, the memory limit for Java is sufficient for most ODS Graphics applications. However, some tasks are very memory intensive and might exhaust all available Java memory, resulting in an OutOfMemoryError condition. You might encounter Java memory limitations in the following cases:

- the product of the output size and the DPI setting results in very large output
- a classification panel has a very large number of classifier crossings
- a scatter plot matrix has a large number of variables
- creating 3-D plots and 2-D contours, which are memory intensive to generate
- a plot has a very large number of marker labels
- a plot uses many character variables or has a large number of GROUP values
- using the SG Editor to edit a graph with a large amount of data
Managing a Java Out of Memory Error

If you encounter a Java OutOfMemoryError, then you can try executing your program again by restarting SAS and specifying a larger amount of memory for Java at SAS invocation.

To determine what the current Java memory settings are, you can submit a PROC OPTIONS statement that shows the value of the JREOPTIONS option:

```sas
proc options option=jreoptions;
run;
```

After you submit this procedure code, a list of JREOPTIONS settings is written to the SAS log. The JREOPTIONS option has many suboptions that configure the SAS Java environment. Many of the suboptions are installation and host specific and should not be modified, especially the ones that provide installed file locations. For managing memory, look for the -Xmx and -Xms suboptions:

```sas
JREOPTIONS=(/* other Java suboptions */ -Xmx128m -Xms128m)
```

- **-Xms**
  Use this option to set the minimum Java memory (heap) size, in bytes. Set this value to a multiple of 1024 greater than 1MB. Append the letter k or K to indicate kilobytes, or m or M to indicate megabytes. The default is 2MB. Examples:

  - -Xms6291456
  - -Xms6144k
  - -Xms6m

- **-Xmx**
  Use this option to set the maximum size, in bytes, of the memory allocation pool. Set this value to a multiple of 1024 greater than 2MB. Append the letter k or K to indicate kilobytes, or m or M to indicate megabytes. The default is 64MB. Examples:

  - -Xmx83886080
  - -Xmx81920k
  - -Xmx80m

As a general rule, you should set the minimum heap size (-Xms) equal to the maximum heap size (-Xmx) to minimize garbage collections.

Typically, SAS sets both -Xms and -Xmx to be about 1/4 of the total available memory or a maximum of 128M. However, you can set a more aggressive maximum memory (heap) size, but it should never be more than 1/2 of physical memory.

You should be aware of the maximum amount of physical memory your computer has available. Let us assume that doubling the Java memory allocation is feasible. So when you start SAS from a system prompt, you can add the following option:

```sas
-jreoptions (-Xmx256m -Xms256m)
```
Alternatively, you might need to specify the setting in quotation marks:

```
-jreoptions '(-Xmx256m -Xms256m)'
```

The exact syntax varies for specifying Java options, depending on your operating system, and the amount of memory that you can allocate varies from system to system. The set of JRE options must be enclosed in parentheses. If you specify multiple JREOPTIONS system options, then SAS appends JRE options to JRE options that are currently defined. Incorrect JRE options are ignored.

If you choose to create a custom configuration file, then you would simply replace the existing -Xms and -Xmx suboption values in the JREOPTIONS=(all Java options) portion of the configuration file.

For more information, see the SAS Companion for your operating system.