

Paper 237-2010

## Creating Presentation-Quality ODS Graphics Output

Dan Heath, SAS Institute Inc., Cary, NC

### ABSTRACT

ODS Graphics is a flexible system that enables you to generate statistical graphics automatically from a number of SAS® products, as well as create your own custom graphics using the SG procedures and the Graph Template Language. When creating your graphics for a presentation or a report, there are a number of tips and techniques you can use to improve the effectiveness of your graphics and convey your information clearly. Areas discussed in this paper include effective graphics principles, style control, resolution control, and presentation medium considerations.

### INTRODUCTION

The production release of the ODS Graphics system in SAS 9.2 was a major advancement in graphics creation. The system enables you to generate statistical graphics automatically or to create your own custom graphics. Even with the advancement in the graphics output, it is still important to understand some principles and techniques for creating effective graphs for your presentations.

Presentations have a variety of forms, including slide presentations, electronic documents, and printed reports. Each of these mediums has its own considerations for creating graphics, particularly in visual style and graphics resolution. This paper discusses these considerations and the options you have to control them. In addition, some general principles for creating effective graphics are discussed.

### IMPORTANCE OF GRAPH CONTENT

It is important that each graph you create convey the intended message clearly and effectively. Naomi Robbins (2005, p.6) described effectiveness by saying, "One graph is more effective than another if its quantitative information can be decoded more quickly or more easily by most observers." This principle is true, regardless of the presentation type. Some content factors that impact a graph's effectiveness include the following:

- Choosing effective chart types for your data
- Using visual attributes that emphasize data
- Using effective labeling of data
- Using uncluttered axes
- Using graph layouts to eliminate clutter

### CHOOSING EFFECTIVE CHART TYPES

The graph type that you choose to represent your data can greatly impact how well your data is interpreted. For example, if you have data with only two or three categories, you might be able to represent the data with a pie chart.

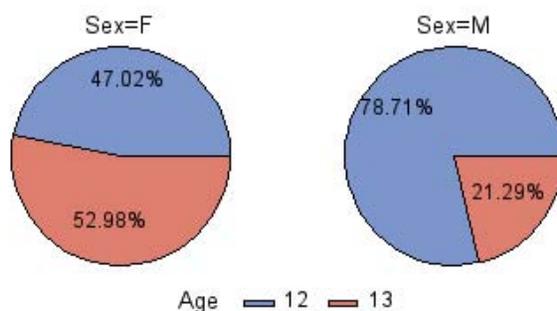


Figure 1. Pie Chart with Two Categories

Even without the percentage labels, you can compare the relative sizes of the two ages to see which is larger. However, if you add more ages, the size comparison becomes a lot more difficult.

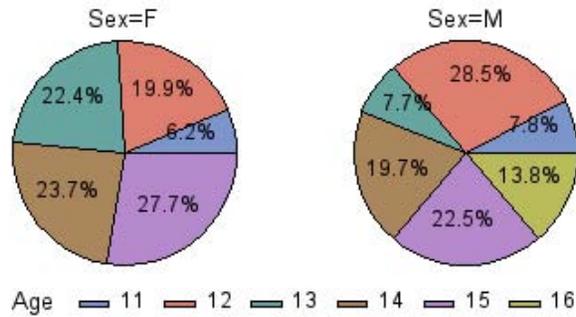
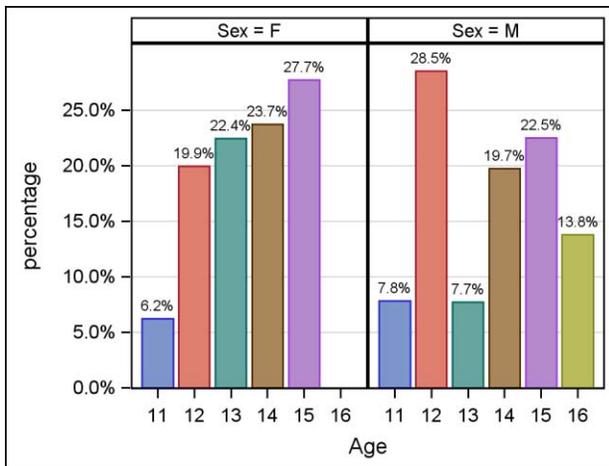


Figure 2. Pie Chart with Multiple Categories

A bar chart might be a more effective chart type for this situation. With a bar chart, you can more easily compare the same age across males and females, and compare all ages in each gender. The comparisons in the bar chart are visually easier because most people are not able to judge angles in pie charts very well (Robbins 2005, p.49).

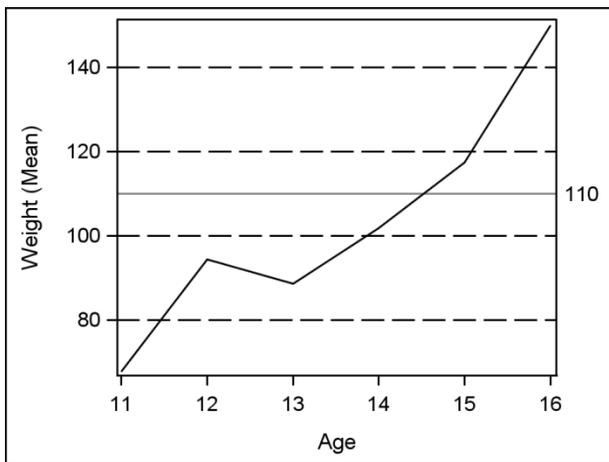


```
proc sgpanel data=percent_data
  noautolegend;
where percentage ne .;
panelby sex;
rowaxis grid;
vbar age / group=age datalabel
  response=percentage
  nostatlabel;
run;
```

Figure 3. Bar Chart with Multiple Categories

**USING VISUAL ATTRIBUTES THAT EMPHASIZE DATA**

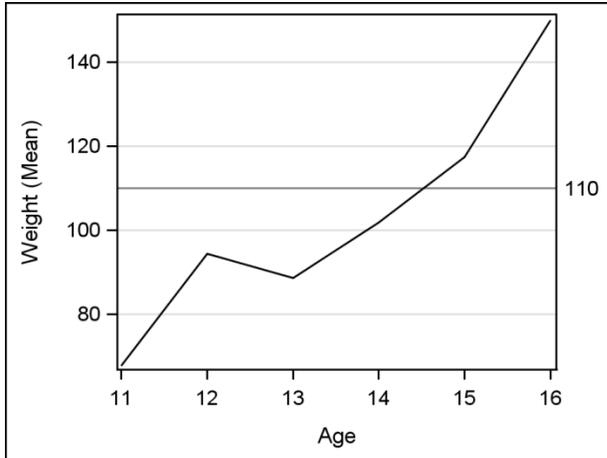
The data in your graph should stand out from any supporting elements, such as grid lines and reference lines (Robbins 2005, p.159-161). In Figure 4, the reference line at 110 and grid lines are the same color. Even though the line patterns are different, it takes effort to see past the visual congestion and read the message in the data.



```
proc template;
define style styles.mylisting;
parent=styles.listing;
class GraphGridLines /
  linestyle = 4
  contrastcolor = cx000000
;
end;run;
ods listing style=mylisting;
proc sgplot data=sashelp.class;
yaxis grid;
refline 110 / label
  lineattrs=(color=cx000000
  pattern=longdash);
vline age / response=weight stat=mean;
run;
```

Figure 4. Congested Line Plot

To improve this graph, the grid lines and reference line should be more subdued in comparison to the line chart. Fortunately, many of the ODS styles in SAS are designed with this concept. If you remove the extra syntax from Figure 4 and apply the LISTING style, you get a much cleaner graph. (See Figure 5.) There might be times that you need to directly control the appearance of grid and reference lines, as in Figure 4. If you do, keep the data emphasis principle in mind.



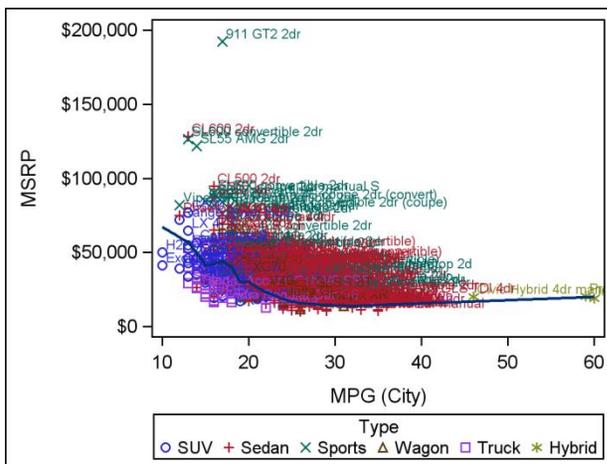
```
ods listing style=listing;
proc sgplot data=sashelp.class;
  yaxis grid;
  refline 110 / label;
  vline age / response=weight stat=mean;
run;
```

Figure 5. Subdued Style for Supporting Elements

## USING EFFECTIVE LABELING

Data labeling is an important way to associate data values with plot features and to add information that is not represented in the graph. Data labeling techniques include directly labeling plot features inside or outside the axis area and using a plot legend.

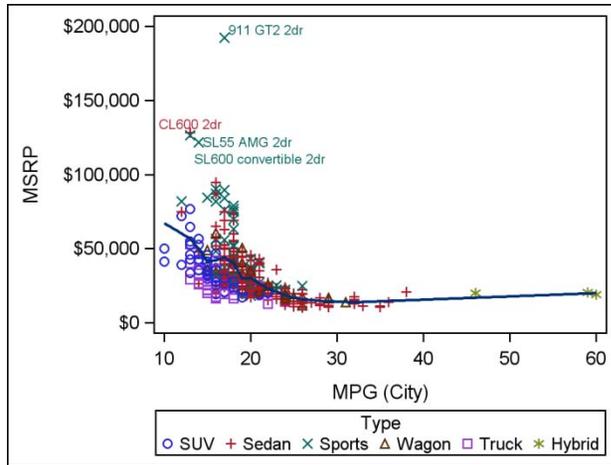
A major consideration when choosing the best type of labeling is the volume of data that is displayed. In Figure 6, 428 scatter points are displayed with corresponding labels. The labeling creates clutter in the graph and makes it difficult to see the patterns in the data.



```
proc sgplot data=sashelp.cars;
  scatter x=mpg_city y=msrp /
    group=type datalabel=model;
  loess x=mpg_city y=msrp / nomarkers;
run;
```

Figure 6. Loess Plot with Labels

However, for plots with a lot of points, data labeling can be used to label outlying points of interest. In Figure 7, the points with an MSRP of \$100,000 or more are labeled with the model names of the automobiles. To label specific points on a plot, create a new column with the label information for only those points that meet your labeling criteria. For this example, a DATA step was used to copy the model name into a column called EXPENSIVE when the MSRP was greater than or equal to \$100,000. Only the nonmissing values in the EXPENSIVE column are drawn by the ODS Graphics system.



```
data cars;
  set sashelp.cars;
  if (msrp >= 100000) then
    expensive=model;
run;
proc sgplot data=sashelp.cars;
  scatter x=mpg_city y=msrp /
    group=type datalabel=expensive;
  loess x=mpg_city y=msrp / nomarkers;
run;
```

Figure 7. Labeling Specific Scatter Points

Figure 8 shows an example of the effective use of data labeling with series plots. When possible, direct labeling of plot lines is preferred over putting them in a legend (Robbins 2005, p.213). The series plots are simple so they can be labeled directly without causing clutter in the graph.

```
title "CL Bundle Compliance & CL Infection Rate";
proc sgplot data=compliance;
  series y=Compliance x=month / markers datalabel=Comp_label
    curvelabel=" Compliance" curvelabelloc=inside;
  series y=BSI x=month / y2axis markers datalabel=BSI_label
    curvelabel=" BSI" curvelabelloc=inside;
  yaxis label="% Compliance" values=(0 to 1 by 0.2) grid;
  y2axis label="BSI per 1000 CL Days" values=(0 to 20 by 4);
run;
```

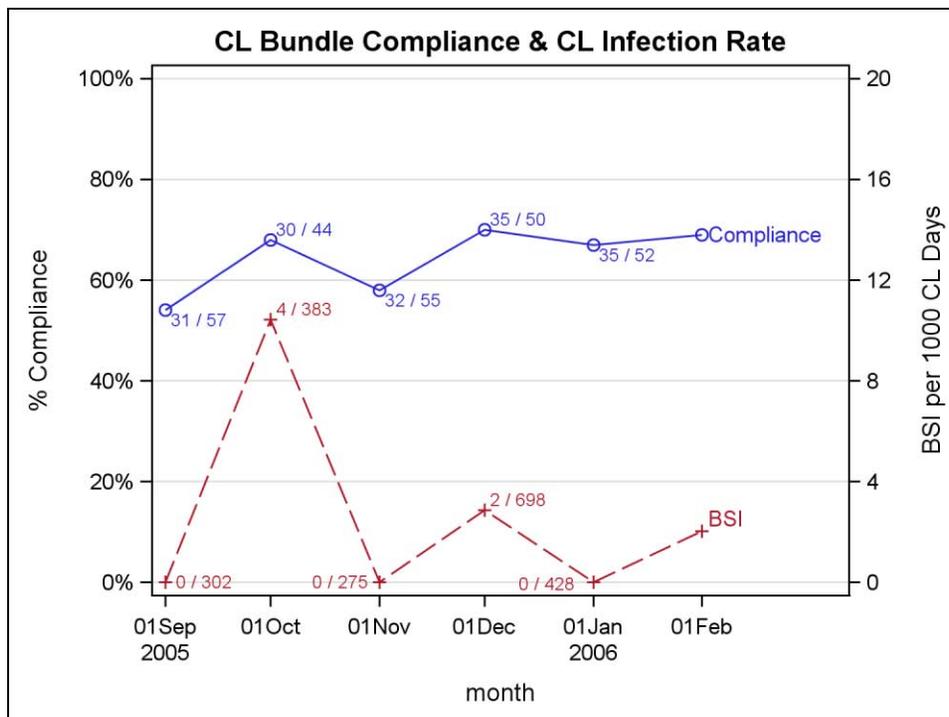
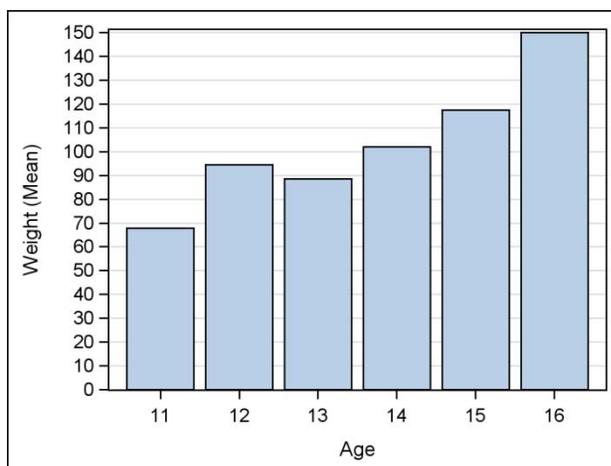


Figure 8. Series Plot with Data Labels and Curve Labels

## USING UNCLUTTERED AXES

An axis with too many tick marks can actually detract from the appearance of a graph (Robbins 2005, p.183). In Figure 9, the Y-axis is set to show increments of 10. That level of detail is not necessary to decode the graph and might be distracting.



```
proc sgplot data=sashelp.class;
  yaxis values=(0 to 150 by 10) grid;
  vbar age / response=weight stat=mean;
run;
```

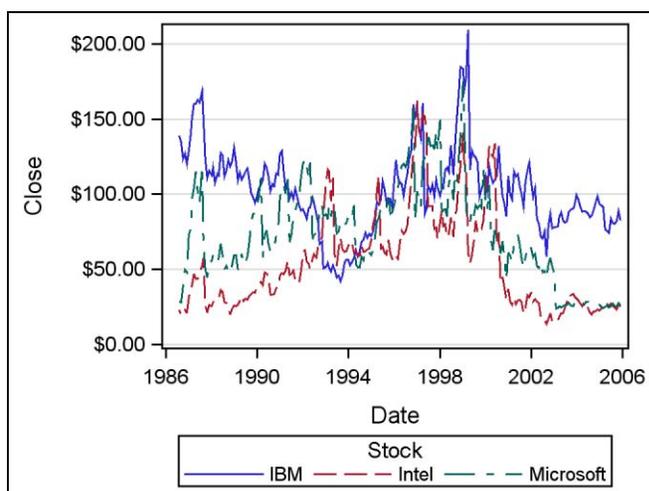
Figure 9. Y-Axis with Excessive Tick Marks

The axis generation of the ODS Graphics system is designed to automatically generate just enough tick marks to interpret the data. In addition, internal algorithms prevent the data from getting “squeezed,” which can occur when data is forced to be plotted between two “nice” tick marks. Whether tick marks are generated automatically, or you set the tick marks yourself, it is important to create an axis that balances the decoding of the graph with its visual impact.

## USING GRAPH LAYOUTS TO ELIMINATE CLUTTER

A useful technique to compare plots is to overlay them in the same data space. However, there are times when the number of overlays or the variations of the plots create a display that is difficult to decode.

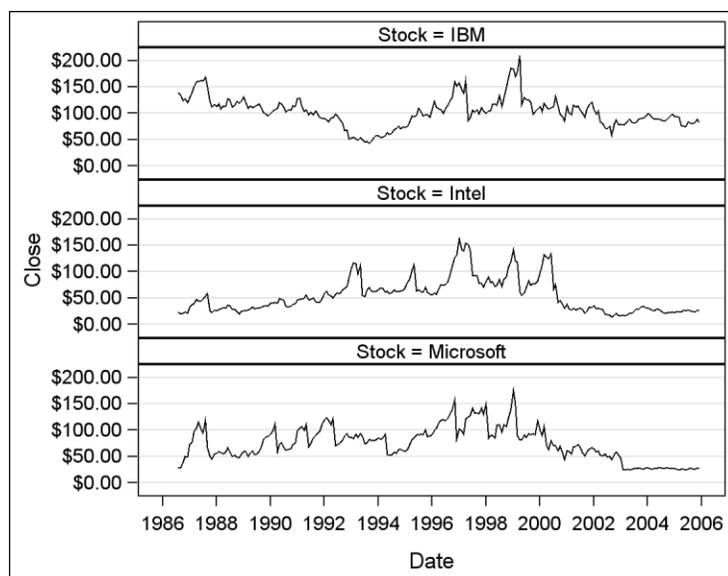
Consider the example in Figure 10. This graph compares the price history of three stocks by plotting their closing prices over time in the same data space. The clutter in the graph makes it difficult to see the general trend of each stock.



```
proc sgplot data=sashelp.stocks;
  yaxis grid;
  series x=date y=close /
    group=stock;
run;
```

Figure 10. Overlaid Series Plot

An alternative way to view this data is to split each stock trend into its own cell (Naomi 2005, p.172-173). Instead of using the STOCK variable as a grouping variable for the SERIES plot, use it as a class variable to create unique cells for each stock. (See Figure 11.) Now, the trend of each stock is easier to decode. And, the uniform axes enable you to make comparisons over time.



```
proc sgpanel data=sashelp.stocks;
  panelby stock / columns=1;
  rowaxis grid;
  series x=date y=close;
run;
```

Figure 11. Series Plot in Each Cell

## GLOBAL OPTIONS

The ODS GRAPHICS statement contains global options that affect all output from the ODS Graphics system. The following sections highlight a few of the options that can affect the size and content of your graph.

### ANTIALIAS / ANTIALIASMAX

Antialiasing is a technique for smoothing jagged lines in your graph, including the lines used to draw markers. The ODS GRAPHICS statement has two options for controlling antialiasing: ANTIALIAS and ANTIALIASMAX. The ANTIALIAS option turns antialiasing on or off. (The default value is ON.) The ANTIALIASMAX option sets a threshold to turn off antialiasing when the data has a specified number of observations. (The default value is 600.) Antialiasing is disabled based on the number of observations because antialiasing uses a lot of resources. And, drawing a large number of points in most plots would look the same without antialiasing. On the other hand, some plots might have more than 600 observations, but they are plotted in a way that would benefit from antialiasing. For these plots, the threshold can be increased.

When the threshold is met, a descriptive note similar to the following is generated in the SAS log:

```
NOTE: Marker and line antialiasing has been disabled because
      the threshold has been reached. You can set
      ANTIALIASMAX=700 in the ODS GRAPHICS statement to restore
      antialiasing.
```

The note not only tells you that the threshold has been met, but also the observation count to set on the ANTIALIASMAX option to restore antialiasing.

### BORDER

The BORDER option turns the border on or off around the entire graph. The default value is ON.

### MAXLEGENDAREA

The MAXLEGENDAREA option limits the amount (in a percentage) of graph area that a legend can use. (The default value is 20.) This option prevents legends from dominating the graph area. However, if you require a legend to decode your graph, and your legend gets dropped, you can either increase the value of MAXLEGENDAREA or increase the height or width of the graph.

### SCALE

The SCALE option enables or disables the scaling of fonts and markers. (The default value is ON.) Scaling is based on the DESIGNHEIGHT of the graph (480 pixels by default). Graphs that are larger or smaller than this DESIGNHEIGHT scale fonts and markers. If you do not want to scale the fonts and markers (perhaps because of publication requirements), you should set the SCALE option to OFF.

## WIDTH / HEIGHT

The WIDTH and HEIGHT options change the size of the graph. The dimensions can be specified in units supported by ODS, including SPX (special pixels), PX (pixels), IN (inches), CM (centimeters), and MM (millimeters). The default unit is SPX. It is recommended that you always provide a unit such as PX, IN, CM, or MM with the dimension value (for example, 480px).

The ODS Graphics system has a concept of DESIGNWIDTH and DESIGNHEIGHT. These Graph Template Language options control the layout dimensions of the graph. (The SG procedures use the default values.) The default DESIGNWIDTH is 640px, and the default DESIGNHEIGHT is 480px. If you specify a width or height different from the corresponding DESIGNWIDTH or DESIGNHEIGHT, scaling occurs in the graph. (See the SCALE option above.)

The WIDTH and HEIGHT options can be specified together or independently. If only one is specified, then the other option is automatically adjusted to maintain the aspect of the graph. To change the aspect, you must specify both options.

## STYLE ISSUES

Styles are an important consideration when creating your presentation. Beyond using data emphasis, other style considerations specifically related to the medium of your presentation can be used. These considerations include font size, line thickness, marker size, and color (or black and white). The following sections discuss these considerations.

### SLIDE PRESENTATIONS

When creating graphics for a slide presentation, one of the biggest considerations is the number of attendees. If the presentation is for a large group, you need to make sure that your graphics can be clearly seen by the people in the back of the room. This usually means increasing the size of certain graph features, such as graph fonts, plot lines, and plot markers. Increasing the font size helps you read the text and axis information in the graph. Increasing plot line thickness and marker size helps you differentiate between plots in the graph.

Line thickness is particularly important when the only distinguishing attribute between plots is color (for example, there are no line patterns). As lines get thinner, lines with similar colors become harder to distinguish. Markers have a similar problem. As markers with similar colors get smaller, their colors become harder to distinguish, particularly if the marker is not filled. Marker shapes become harder to distinguish as markers get smaller. Different marker shapes tend to deteriorate into similar-looking shapes, which make it harder to distinguish one marker from another.

There are a few techniques for increasing the size of the graph contents. The first technique involves a creative use of the graph WIDTH and HEIGHT options to change the graph resolution. (See the section on graph resolution later in the paper.) By setting the width and height of the graph to be *smaller* than you need, and also setting a high resolution, you can import the graph into your presentation and resize the graph to the appropriate size. This type of resizing causes everything in the graph to increase in size, while maintaining a high-quality graphic.

The second technique involves modifying an ODS style. ODS styles are generally designed for the computer screen or printed output. However, these styles can easily be enhanced to derive a style for a large-audience slide presentation.

The first step in this technique is to choose the ODS style with the appropriate appearance. The following code contains the minimal code to create a new ODS style from an existing ODS style. For this example, a new style called PRESENTATION is created and the style attributes are inherited from the LISTING style. If this code were submitted using the PRESENTATION style, the output would look identical to the output using the LISTING style.

```
proc template;
  define style styles.presentation;
    parent = styles.listing;
  end;
run;
```

To enlarge the graph fonts and plot features of the PRESENTATION style, you need to make changes to two style elements: GRAPHFONTS and GRAPHDATADEFAULT. To get these elements from the LISTING style, you have two choices:

1. Run the following code to write the style to the SAS log:

```
proc template;
  source styles.listing;
run;
```

2. Perform the following steps to open the style in an editor in a DMS session:

- a. Type `odst` in the command field.
- b. Expand **SASHELP.TMPLMST**.
- c. Click the **Styles** folder.
- d. Double-click the **LISTING** style.

When you look at the LISTING style, you will notice that the style definition is actually small. This is because the LISTING style inherits most of its content from the DEFAULT style. (See the PARENT option.) You will also notice that the GRAPHDATADEFAULT element is not in the LISTING style. Therefore, you need to get that element from the DEFAULT style using the same steps as for the LISTING style.

Copy the two elements into the new PRESENTATION style definition. The result should look something like the following:

```
proc template;
  define style styles.presentation;
    parent = styles.listing;
    style GraphFonts "Fonts used in graph styles" /
      'GraphDataFont' = (", <MTsans-serif>", 7pt)
      'GraphUnicodeFont' = ("", 9pt)
      'GraphValueFont' = (", <MTsans-serif>", 9pt)
      'GraphLabelFont' = (", <MTsans-serif>", 10pt)
      'GraphFootnoteFont' = (", <MTsans-serif>", 10pt)
      'GraphTitleFont' = (", <MTsans-serif>", 11pt, bold)
      'GraphAnnoFont' = (", <MTsans-serif>", 10pt);
    class GraphDataDefault /
      endcolor = GraphColors('gramp3cend')
      neutralcolor = GraphColors('gramp3cneutral')
      startcolor = GraphColors('gramp3cstart')
      markersize = 7px
      markersymbol = "circle"
      linethickness = 1px
      linestyle = 1
      contrastcolor = GraphColors('gdata')
      color = GraphColors('gdata');
  end;
run;
```

The goal for the new PRESENTATION style is to make the fonts bigger, the plot lines thicker, and the marker sizes bigger. To make the fonts bigger, increase the point sizes for each font in GRAPHFONTS. To make the plot lines thicker, increase the number of pixels for the LINETHICKNESS attribute. To make the marker sizes bigger, increase the number of pixels for the MARKERSIZE attribute. After you make these changes, submit the code to store the template in SASUSER.TEMPLAT. You can now use the PRESENTATION style by using the STYLE option in the ODS DESTINATION statement.

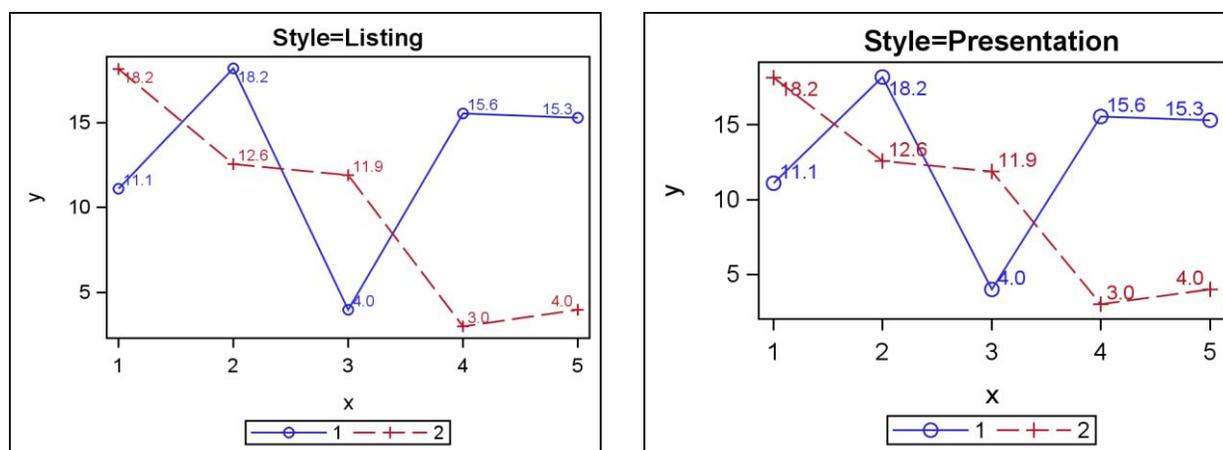


Figure 11. LISTING Style Versus PRESENTATION Style

Figure 11 demonstrates the effect of these changes. For this example, the fonts were increased by 2 points, the plot line thickness was set to 2 pixels, and the marker size was set to 9 pixels. Varying both the line color and the line style improves the differentiation of the plot lines. However, if you want your style to contain only solid plot lines, you can copy the GRAPHDATA1 through GRAPHDATA12 style elements into your style, and set the LINESTYLE attribute to 1 in all of these elements.

## PRINTED REPORTS

ODS styles are generally designed for the computer screen or printed output. When generating graphics for printed output, one major consideration is whether the final output is in color or in black and white. If the final output is in black and white, you should not generate your graphics using a color ODS style. Colors in your graphs with the same lightness and saturation will appear to be the same color when printed in black and white, even if they are different hues. Because the color ODS styles are not designed with these black and white translation considerations, you could get output that is difficult to decode.

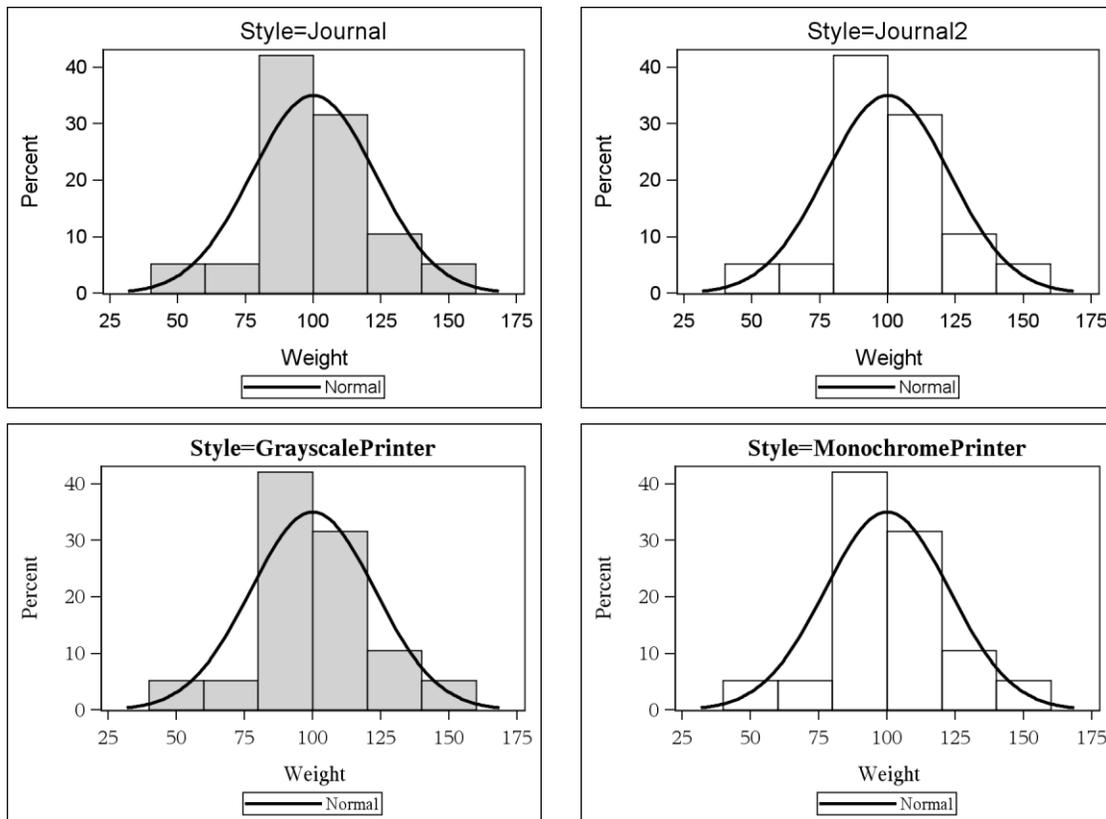


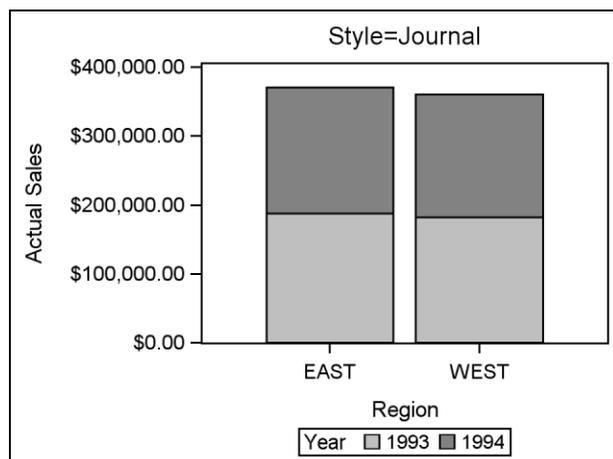
Figure 12. Histograms Using Gray Scale and Monochrome Styles

The solution is to use one of the black and white or gray scale styles defined in ODS. These styles are designed to provide distinction between plots without using color. Figure 12 shows an example of histogram output using each of these styles. The key difference in the output between the JOURNAL styles and the PRINTER styles is the font definitions. The JOURNAL styles use sans-serif fonts (such as Arial), while the PRINTER styles use serif fonts (such as Times).

The plot line and marker color for all of these styles is black. Line style and marker shape attributes are used for plot differentiation. For gray scale styles, plots with filled areas are differentiated by shades of gray. For monochrome styles, most plots that contain areas are unfilled, giving a minimal ink appearance. These plots include box plots, ellipses, band plots, and histograms. Bar charts are an exception because grouped bars need to be differentiable. However, bar charts can be set to have this appearance through syntax options in the Graph Template Language and in the SG procedures.

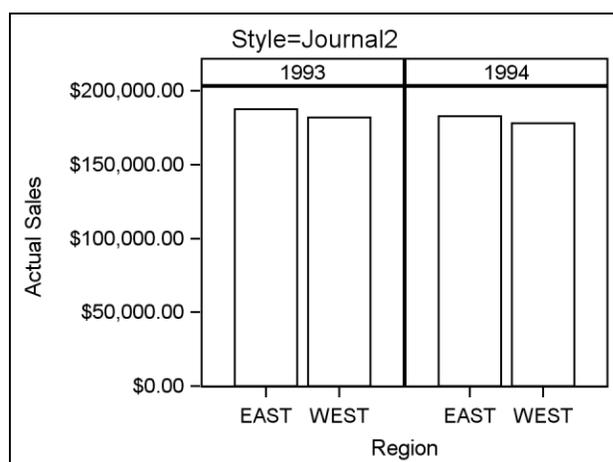
Choosing a gray scale style versus a monochrome style depends on your graph content and the requirements of your publisher. For plots with no filled areas, gray scale and monochrome styles are graphically equivalent. For most plots with filled areas, the choice is a matter of aesthetics or publishing requirements. For a grouped bar chart like the one

in Figure 13, gray scale is necessary to differentiate the groups. Another approach for rendering this group variable is to use the variable in a panel instead of in a bar chart, making the fill color unnecessary. (See Figure 14.)



```
title "Style=journal";
proc sgplot data=sashelp.prdsale;
vbar region / response=actual group=year;
run;
```

Figure 13. Gray Scale Grouped Bar Chart



```
title "Style=journal2";
proc sgpanel data=sashelp.prdsale;
panelby year / novarname;
vbar region / response=actual nofill;
run;
```

Figure 14. Monochrome Paneled Bar Chart

## GRAPHICS RESOLUTION

Regardless of your presentation medium, the resolution of your graphics is an important consideration. Resolution is usually measured in dots per inch (dpi). The higher the dpi value, the higher the image resolution of your graph. You would think that you should always set a very high dpi value for your image output, but that is not always the case. Some applications do not handle high-resolution images well. In addition, it takes more time and memory resources to generate high-resolution images, and more disk space to store them. Therefore, before you automatically choose a high dpi, you should consider the medium for your presentation.

## WEB PAGE RESOLUTION

One medium where high-resolution images tend not to work well is a Web page. The graphs in Figure 15 are screen captures from Internet Explorer. Notice that the 200dpi image appears to have rendering artifacts when compared to the 100dpi image. However, if you increase the zoom level of the browser display to 200%, the 200dpi image looks much better. Most people do not increase the zoom level on the Web page they are viewing. Therefore, you want the best-looking graph at the 100% zoom level, which is 100dpi in this case. Some browsers handle high dpi images better than others, but you want your output to look good regardless of the browser. That is why the default image resolution for ODS HTML is 100dpi.

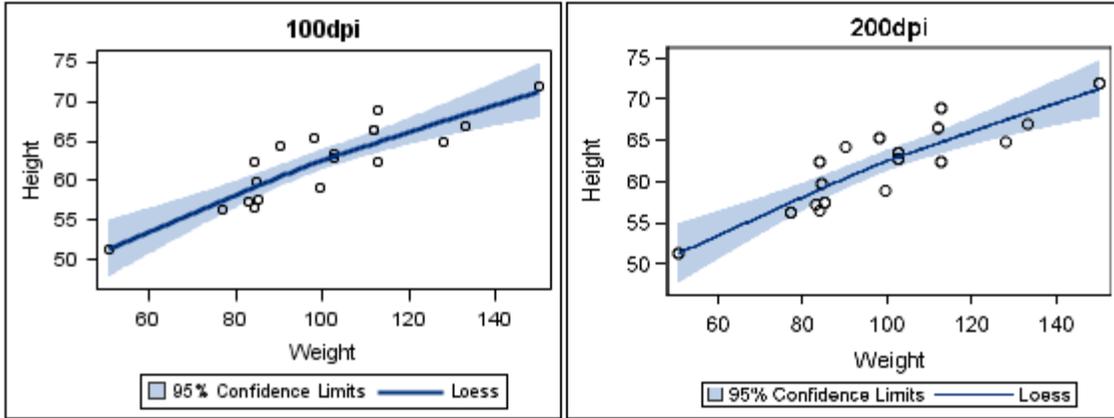
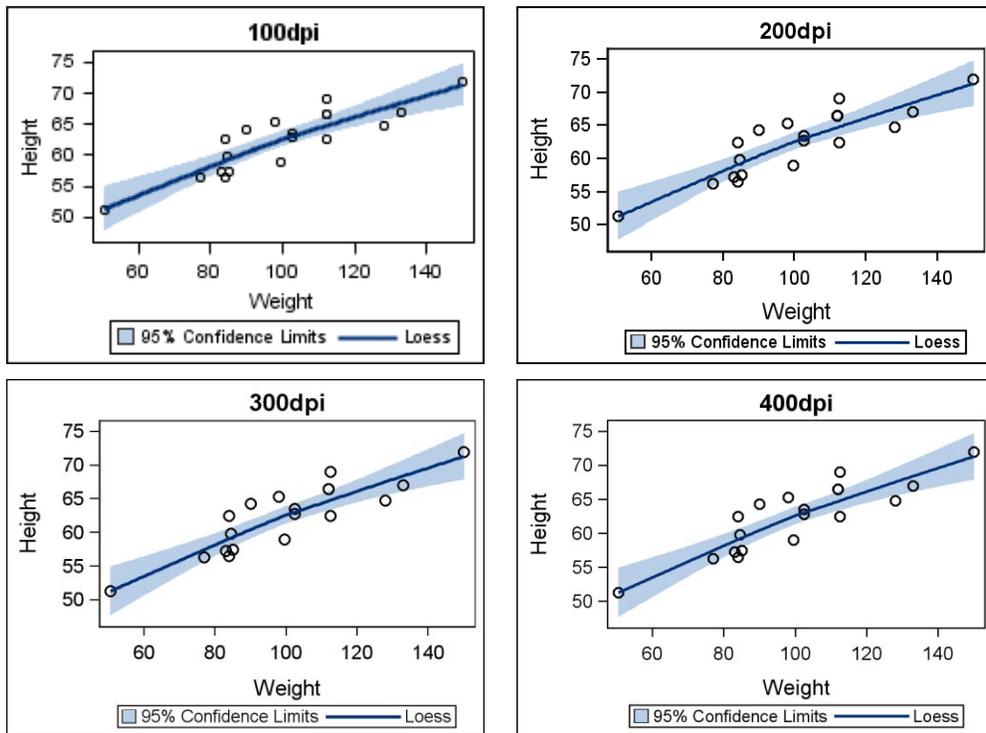


Figure 15. Screen Capture from Internet Explorer

**SLIDE PRESENTATION RESOLUTION**

To create high-quality graphics for your slide presentation, you should choose a resolution higher than 100dpi. However, you might be surprised to know that a very high resolution is not required to create graphics suitable for a slide presentation.



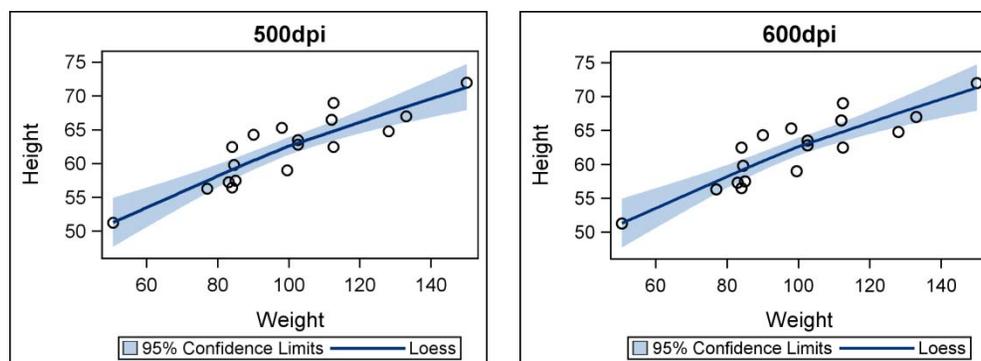


Figure 16. Loess Fit at Different Resolutions

Compare the Loess fit plots in Figure 16. Notice that increasing the resolution from 100dpi to 200dpi gives you a significant improvement in quality, particularly for the fonts. As you go beyond 200dpi, the improvements become more marginal. Because high dpi images take more time and memory to produce and more disk space to store them, you might want to produce your graphics using 200dpi or 300dpi.

### PRINTED REPORT RESOLUTION

Graphics for printed reports can be the most demanding in terms of resolution. Printer resolutions usually start at 150dpi, and the dpi value doubles as you increase printer quality. Most printers today support 600dpi to 1200dpi. (Some high-end printers even support 2400dpi.) Generally, you want your image dpi to match your printer dpi to prevent any interpolation artifacts. If you cannot produce an image large enough to match the printer dpi, make sure you choose a lower dpi that is correct for the printer (75, 150, 300, 600, or 1200) so that interpolation occurs smoothly. This rule is also true for PDF document generation, even if it is never printed.

When generating high dpi images in the ODS Graphics system, it might be necessary to adjust the memory resources used by the system. The system uses Java technology to render the actual graphics. The Java Virtual Machine (JVM) is initialized with a set amount of memory at startup. Different factors impact how fast that memory is consumed, such as large data sets, large image sizes, and image dpi. The default memory allocation is large enough for most situations. However, if all factors are large enough, the system might complain about being out of memory.

Fortunately, you can change the setting of JVM memory in SAS to handle these situations. In the SAS configuration file, there is an entry for JREOPTIONS that looks similar to the one below. The -XXM and -XMS options control memory allocation. The -XXM option specifies how much memory to allocate, and the -XMS option specifies how much of that memory to initialize at startup. For this specific SAS configuration file, the JVM is allocated 128 MB of memory. If you increase the memory allocation, you should make the increase in increments of 32 MB. Make sure that the number you specify is no more than half of the physical memory in your computer. For example, if your computer has 1 GB of memory, make sure that the memory allocation does not exceed 512 MB.

```

/* Options used when SAS is accessing a JVM for JNI processing */
-JREOPTIONS=(-Dsas.jre.libjvm=C:\PROGRA~1\Java\JRE15~1.0_1\bin\client\jvm.dll
-Djava.security.policy=!SASROOT\core\sasmisc\sas.policy
-Dsas.ext.config=!SASROOT\core\sasmisc\sas.java.ext.config
-
Dsas.app.class.path=C:\PROGRA~1\SAS\SASVER~1\9.2\eclipse\plugins\tkjava.jar
-DPFS_TEMPLATE=!SASROOT\core\sasmisc\qrpfstpt.xml
-
Djava.class.path=C:\PROGRA~1\SAS\SASVER~1\9.2\eclipse\plugins\SASLAU~1.JAR
-Djava.system.class.loader=com.sas.app.AppClassLoader
-XXm128m
-Xms128m
-Djava.security.auth.login.config=!SASROOT\core\sasmisc\sas.login.config
-Dtkj.app.launch.config=!SASROOT\picklist)

```

## CONCLUSION

The ODS Graphics system is a flexible system for creating graphics. There are a large number of layouts, plot types, and options that can be combined in many creative ways to make the presentation-quality graphics you need. You should continue to explore this functionality by looking at the *SAS/GRAPH Graph Template Language User's Guide* and the *SAS/GRAPH Statistical Graphics Procedures Guide*.

## REFERENCES

Robbins, Naomi B. 2005. *Creating More Effective Graphs*. New Jersey: John Wiley & Sons, Inc.

## ACKNOWLEDGMENTS

I want to thank Susan Schwartz, Sanjay Matange, and Melisa Turner for their valuable input.

## RECOMMENDED READING

Robbins, Naomi B. 2005. *Creating More Effective Graphs*. New Jersey: John Wiley & Sons, Inc.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dan Heath  
SAS Institute Inc.  
500 SAS Campus Drive  
Cary, NC 27513  
E-mail: Dan.Heath@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.