

Paper 254-2008

SAS® Graphics on ODS 9.2 Performance-Enhancing Steroids

Daniel O'Connor, SAS Institute Inc, Cary, NC

ABSTRACT

In the midst of performance-enhancing drug scandals in professional sports, you would think that SAS® graphics on ODS 9.2 performance-enhancing steroids would be a bad thing. That is until you experienced the ability to customize professional-looking graphics in seconds with a variety of appearances. ODS graphical style enhancements provide advanced capabilities to unleash your creativity without spending endless hours manipulating options. Choose from a list of predefined styles, or create your own to match your corporate theme. ODS styles are both customizable and extendable, and they provide a powerful way to bring your content to life. Explore these new features in HTML, PDF, RTF, and PostScript documents. Astonish your customers and amaze yourself with SAS® 9.2 ODS enhancements for SAS graphics.

INTRODUCTION

Performance-enhancing substances have been used for thousands of years by societies around the world. Athletes, whether young or old, professional or amateur, all give the same reasons for using steroids. They all have the desire to enhance strength, size, speed, appearance, or aggressiveness in order to excel. Now, let's be clear: steroids do not provide superhero abilities, but simply enhance the individual's natural genetic abilities. In SAS 9.2, we have heavily invested in our production graphics capabilities (natural ability) to significantly enhance the appearance, speed, and ease with which we produce publication ready graphics. Significant enhancements in our core technology will provide a steroid effect on our graphics capabilities to increase our competitive advantage. These enhancements are not limited to SAS/GRAPH®, but also include SAS/STAT®, SAS/ETS®, SAS/QC®, SAS/OR®, and Base SAS® as well. This comprehensive effort includes a complete integration with ODS, new SAS/GRAPH procedures allowing you to quickly create complex statistical displays, and the ability to automatically display the results of our statistical analysis graphically.

ODS STYLES

ODS styles are a powerful mechanism to control the overall appearance of your output. A style definition controls presentation characteristics such as colors, which font is used for a title, or how a column heading is displayed in a table. ODS styles simply are a collection of directives that describes how portions of your output are to be presented.

GRAPHICS SUPPORT ODS STYLES

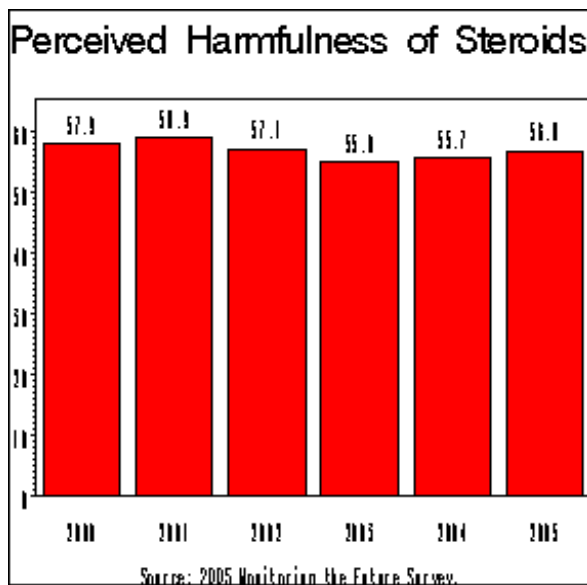
All graphical procedures now support ODS Styles. By default, all colors, fonts, and symbols come from the active style for each open output destination. However, procedures statements, global options (options), and graphics options (GOPTIONS) continue to override the default style settings. Each style has a unique color palette for both tabular and graphical output so that all ODS output has visual continuity. The centralization of stylistic attributes in an ODS style has far-reaching benefits to facilitate rapid code development without tedious attribute manipulation. What this really means is that the quality of your output is significantly better out of the box by just upgrading to SAS 9.2. Here is a very simple example.

Example 1:

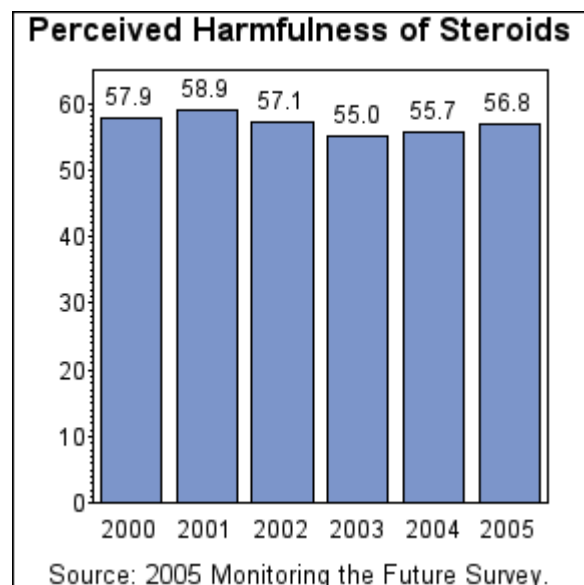
```

data harmful;
input year 4. @6 response 4.1;
output;
datalines;
2000 57.9
2001 58.9
2002 57.1
2003 55.0
2004 55.7
2005 56.8
;
options device=png hsize=3in vsize=3in border;
title "Perceived Harmfulness of Steroids";
footnote "Source: 2005 Monitoring the Future Survey.";
axis1 label=none;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
discrete maxis=axis1 raxis=axis1;
;run;quit;

```



SAS 9.1



SAS 9.2

NEW GSTYLE OPTION

A new SAS option has been added to SAS 9.2 that controls backward compatibility for graphic output.

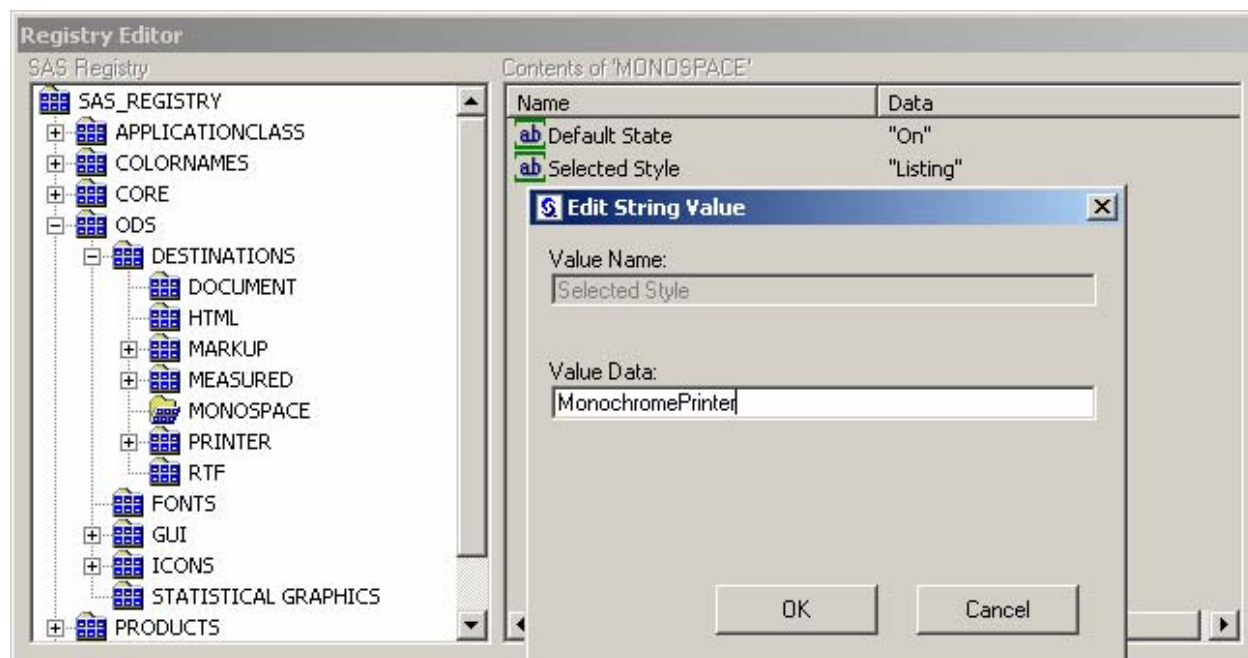
OPTIONS < GSTYLE | NOGSTYLE >;

The default value is GSTYLE which means that graphic output will automatically honor the default style for each open output destination. This means that your output may look different as soon as you upgrade to SAS 9.2. There also is a possibility that this may introduce undesirable side effects in your document if you have modified colors, and fonts when they collide with the style definitions. For complete backward compatibility with SAS 9.1, you should use the NOGSTYLE option.

DEFAULT STYLES

Whether you are preparing HTML for the Web, a PDF for printing, or an RTF for Microsoft Office integration, each third-party format has unique default expectations. Because output destinations serve different purposes each has its own default style. ODS stores the default styles in a customer modifiable hierarchical format called the SAS® Registry. SAS Registry modifications remain in effect from one session to the next and are stored in your SASUSER by default. You may permanently change SAS Registry settings by selecting Solutions > Accessories > Registry Editor PMENU. To change the default style for the Listing destination, perform the following actions:

- 1 Expand the “ODS” tree item.
- 2 Expand the “Destinations” tree item.
- 3 Select the “Monospace” tree item.
- 4 Click the right mouse button and select “Modify” (or Double-click) on the “Selected Style” list item.
- 5 Change the value to an available style (e.g., MonochromePrinter).



The following table lists the default styles for the most common output destinations.

Output Destination	Default Style
Listing	Listing (Graphics ONLY)
HTML	Default
RTF	RTf
Measured RTF	RTF
PDF	Printer
Postscript	Printer
PCL	Printer
Printer	Printer

CHANGING STYLES

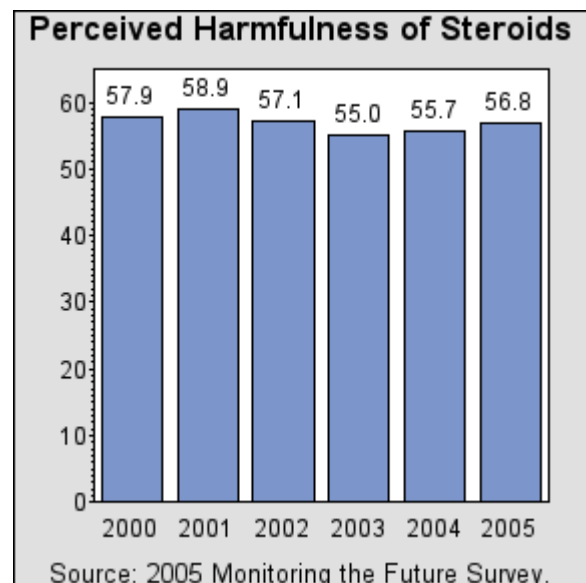
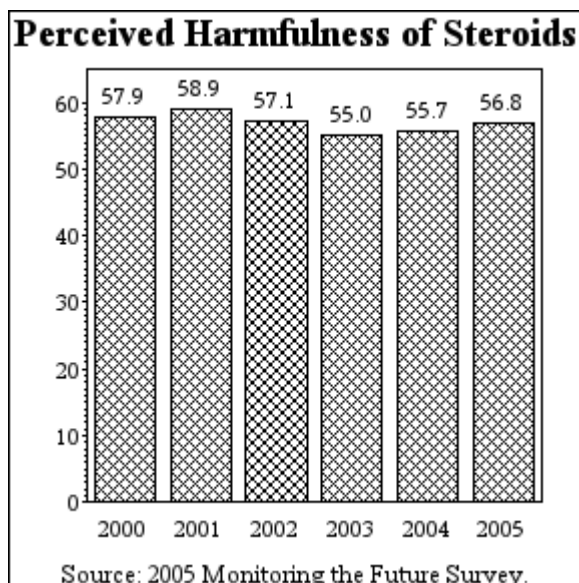
Permanently changing the style is most often not desirable. Normally, you want to change the style only on a temporary basis; say, for the execution of a particular report or for a specific output destination. This can be accomplished by specifying the desired style on the ODS destination statement.

Example 2:

```

options device=png hsize=3in vsize=3in border;
title "Perceived Harmfulness of Steroids";
footnote "Source: 2005 Monitoring the Future Survey.";
axis1 label=none;
ods listing style=MonochromePrinter;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
    discrete maxis=axis1 raxis=axis1;
;run;quit;

```



CREATING YOUR OWN STYLE

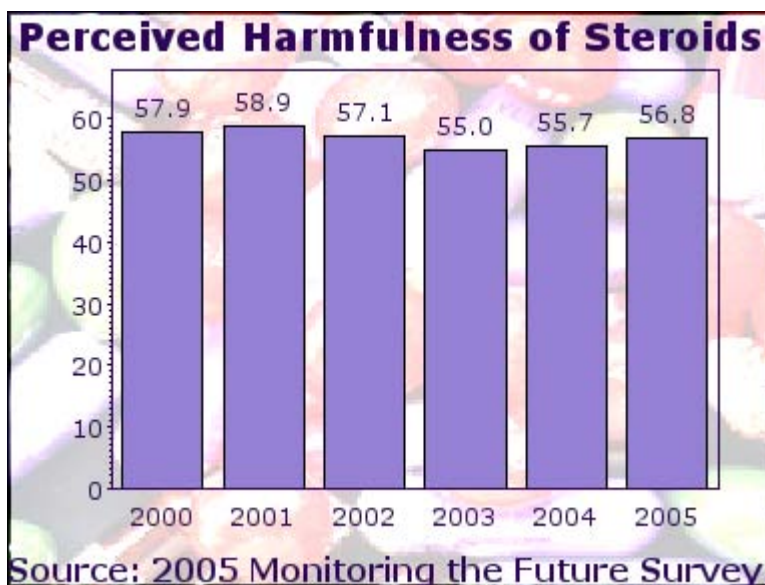
By Default, SAS is shipped with over 50 different styles, but corporations often have sophisticated standards for publishing content. You can modify an existing style or create a completely new style definition from scratch to produce publication-ready content. A few simple style modifications can have dramatic effect on your output.

Seeing that we are focusing on graphs, let's talk about the different elements that make up a graph. A graph can be comprised of any of the following elements: axis, labels, grid lines, background, legends, borders, titles, footnotes, bars, lines, and symbols. Any one of the elements can have a collection of colors, fonts, and a variety of other attributes. ODS styles are organized in a very intuitive manner around the natural graphical elements of a graph. For example, if I wanted to add an image to the background of my graph, I would modify the "GraphBackground" element and provide an image for the "backgroundimage" attribute.

Example 3:

```
proc template;
define style Styles.SGFIImage;
parent=Styles.science;
style GraphBackground
"Graph background attributes" /
backgroundimage = "//dntsrc/u/sasdp0/SGF2008/pillsLighten4.jpg";
end;
run;
```

```
ods listing style=SGFIImage;
Goptions device=png hsize=4in vsize=3in border;
title "Perceived Harmfulness of Steroids";
footnote "Source: 2005 Monitoring the Future Survey.";
axis1 label=none;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
discrete maxis=axis1 raxis=axis1;
;run;quit;
```



As you can see, I chose to extend an existing SAS-supplied style by specifying "parent=Styles.Science". This allows me to inherit all the element definitions in the "Science" definition, and modify only the specific element that I want to modify.

Graphical elements fall into two unique categories, those that are descriptive in nature and those that are data-related. We can think of items like axes, labels, titles, footnotes, and legends as all descriptive in nature. While things like bars, pies, box plot boxes, lines, and symbols as data-related. Let's explore a more sophisticated example that uses gradient colors instead of solid colors. Again, I am going to modify the "GraphBackground" and "GraphData[1..12]" elements, but this time I am going to provide values for the "startcolor", "endcolor", and "gradient_direction" attributes.

Example 4:

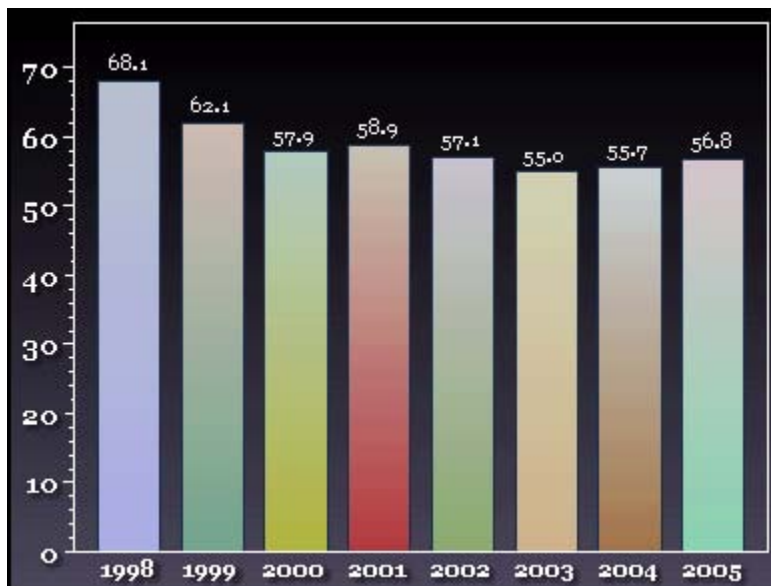
```

data harmful;
input year 4. @6 response 4.1;
output;
datalines;
1998 68.1
1999 62.1
2000 57.9
2001 58.9
2002 57.1
2003 55.0
2004 55.7
2005 56.8
;

proc template;
define style styles.SGFGradient;
parent=styles.education;
style GraphBackground
"Graph background attributes" /
startcolor = cx4a495a endcolor = cx000000 gradient_direction = "YAxis";
style GraphData1 from GraphData1 /
startcolor=GraphColors("gcdata1") endcolor=cxd6deef gradient_direction="yaxis";
style GraphData2 from GraphData2 /
startcolor=GraphColors("gcdata2")endcolor=cxefd6ce gradient_direction="yaxis";
style GraphData3 from GraphData3 /
startcolor=GraphColors("gcdata3")endcolor=cxcee7de gradient_direction="yaxis";
style GraphData4 from GraphData4 /
startcolor=GraphColors("gcdata4") endcolor=cxe7dece gradient_direction="yaxis";
style GraphData5 from GraphData5 /
startcolor=GraphColors("gcdata5") endcolor=cxe7deef gradient_direction="yaxis";
style GraphData6 from GraphData6 /
startcolor=GraphColors("gcdata6") endcolor=cxefefce gradient_direction="yaxis";
style GraphData7 from GraphData7 /
startcolor=GraphColors("gcdata7") endcolor=cxe7eff7 gradient_direction="yaxis";
style GraphData8 from GraphData8 /
startcolor=GraphColors("gcdata8") endcolor=cxf7dee7 gradient_direction="yaxis";
end;

ods html style=SGFGradient gpath="//dntsrc/u/sasdp/SGF2008" gtitle gfootnote;
goptions dev=actximg hsize=4in vsize=3in border;
title "Perceived Harmfulness of Steroids";
footnote "Source: 2005 Monitoring the Future Survey.";
axis1 label=none;
proc gchart;
vbar year / width=10 outside=sum sumvar=response
discrete maxis=axis1 raxis=axis1 pattid=midpoint;
;run;quit;
ods html close;

```



FREE TYPE FONTS ENGINE

Before SAS 9.2, the consistent use of fonts was a real problem when sharing documents across systems, or even between individuals' PCs. Each operating system had a unique set of fonts, so we could never rely on any level of consistency when sharing documents. Often, a document was created with a font that was not available when the document was viewed, and the substituted font compromised the integrity of the document's content. In SAS 9.2, SAS has purchased single-byte as well as double-byte libraries of TrueType fonts that are available for re-distribution. These TrueType fonts will be installed automatically on your system when SAS is installed. We are addressing not only English, but also all the National Language Support requirements in order to further globalize our efforts in this area. There are numerous benefits to using TrueType fonts besides portability. TrueType fonts will also offer quicker rendering, easier access to symbols, international characters, as well as providing an enhanced quality overall. The following tables describe the SAS-supplied single-byte fonts for English, Italian, French, and German languages as well as double-byte fonts for Chinese, Japanese, and Korean.

Single-Byte Character Fonts:

SAS TrueType Font	Type Face	Metric Compatibility
Albany AMT	sans-serif	Arial
Thorndale AMT	serif	Times New Roman
Cumberland AMT	fixed serif	Courier New
Symbol MT	Symbol	Symbol
Monotype Sorts	Symbol	Wingding

Double-Byte Character Fonts:

SAS TrueType Font	Type Face	Language
Monotype Sans WT SC	sans-serif	Simplified Chinese
Monotype Sans WT TC	sans-serif	Traditional Chinese
Monotype Sans WT J	sans-serif	Japanese
Monotype Sans WT K	sans-serif	Korean
Thorndale Duospace WT SC	serif	Simplified Chinese
Thorndale Duospace WT TC	serif	Traditional Chinese
Thorndale Duospace WT J	serif	Japanese
Thorndale Duospace WT K	serif	Korean

GRAPHICS AND DEVICES

Procedures that use the traditional SAS/GRAPH graphics library to produce graphic output require a device driver. This encompasses such products as SAS/GRAPH, SAS/STAT, SAS/OR, SAS/QC, and Base SAS. In SAS 9.2, ODS now automatically selects a default device for each open output destination in a fashion similar to the way that it selects a default ODS style. Not all third-party document types support the same graphical image types, and therefore ODS will optimally select the best graphics device for your document type. This will ensure that the appearance of your graphic output will be presented with the highest quality. Device drivers contain many settings that govern the default appearance of the graphical output such as orientation, size, overall resolution (DPI), and (most importantly) the image type.

DEFAULT DEVICE

Once again because each output destination has a unique purpose, it also has its own default device. For example when you are dealing with PDF output, you probably want your graphic output to be displayed in a full-page mode while HTML output you probably want a more concise size so that the graph is not larger than your browsers viewable area. ODS stores the default devices in the SAS Registry similar to the way that default styles are stored. All the same rules that apply to modifying default styles apply to default devices (see previous default style customization). The following table provides a list of the default devices for the most common output destinations.

Listing	Host Specific Display Device (PC - WIN; UNIX - XCOLOR; MVS - Display Device)
HTML	PNG
RTF	SASEMF
Measured RTF	PNG
PDF	SASPRTC (Color Output) *
Postscript	SASPRTC (Color Output) *
PCL	SASPRTM (Monochrome Output) *
Printer	Host Specific Default Printer *
* Does not support changing the Default Device in the SAS Registry.	

CHANGING DEVICES

Changing a device is as simple as changing either a SAS Global option (e.g., Options device=png;) or a graphic option (e.g., Goptions device=jpeg;). In either case, the most recently specified device driver will be used. In SAS 9.2, we have also introduced a new DEVICE= option on the ODS destinations statements.

Example 3:

```
ods html file="test.html" device=jpeg;
```

Providing a device on the ODS statement takes precedence over both the SAS global option and the graphics option. The printer output destinations do not support the DEVICE= option because they derive their graphics context from the active printer settings.

Each output destination has a unique list of image types that are supported. It is very important to distinguish between an image type and a device driver. Commonly used image types are BMP, PNG, JPEG, EMF, and GIF. You will find that various devices like a digital camera may support JPEG and GIF image types, but not BMP or PNG. While some applications such as Microsoft Paint will support all of the above except EMF. The distinction that I am attempting to make here is that ODS destinations have the similar restrictions. The following is a list of supported image types for each output destination.

Output Destinat	Default Device	
Listing	PNG, GIF, BMP, JPEG, PBM, PS, PDF, TIFF, EMF, EPS, DIB, PCL, PICT, WMF, XBM, XPM, SVG	
HTML	PNG, GIF, JPEG, SVG	
RTF	PNG, EMF, JPEG	
Measured RTF	PNG, EMF, JPEG	
PDF	PNG, PS, PDF, PCL, JPEG, GIF	
Postscript	PNG, PS, PDF, PCL, JPEG, GIF	
PCL	PNG, PS, PDF, PCL, JPEG, GIF	

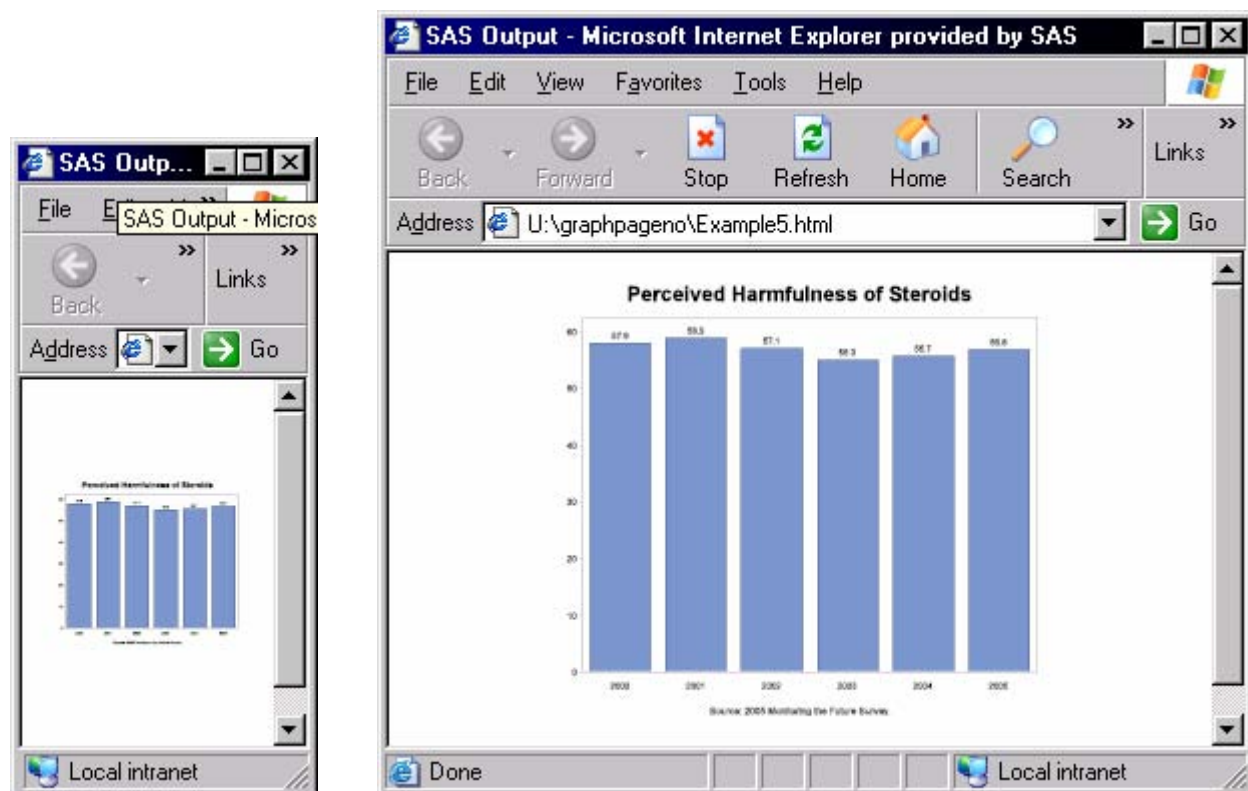
What is the association between a device driver, and an image type? This is quite simply a device driver that knows how to create one and only one image type. For example, the SASEMF device driver creates only an EMF image. Device drivers also have a variety of additional default information such as height, width, resolution, and orientation. So, a device driver is simply a collection of default characteristics that instruct the graph code on how to render the image. Often, there is some confusion surrounding device drivers like PNG that have the same name as the image type that they create. You just have to remember that device drivers contain additional default information that will affect how the graphical output will be rendered.

SCALABLE VECTOR GRAPHICS

The scalable vector graphics (SVG) language is a W3C standard that describes two-dimensional graphics in XML. SAS has introduced this new graphics device driver to produce the SVG language. SVG is not a static image type like PNG, JPEG, and GIF, but a vector graphic format. Vector format graphics can clearly be displayed to any screen resolution and size without compromising the clarity. This makes SVG an ideal format for browsers, computer monitors, PDA, or cell phones. SVG graphics can have additional interactivity capabilities such as tooltips, thumbnails index, zoom, panning, and drill-down capabilities. The following is the same SVG graph; all that I have done is to resize my browser.

Example 5:

```
goptions reset=all;
Goptions device=svg;
ods html file="Example5.html" style=listing;
title "Perceived Harmfulness of Steroids";
footnote "Source: 2005 Monitoring the Future Survey.";
axis1 label=none;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
      discrete maxis=axis1 raxis=axis1;
;run;quit;
ods html close;
```

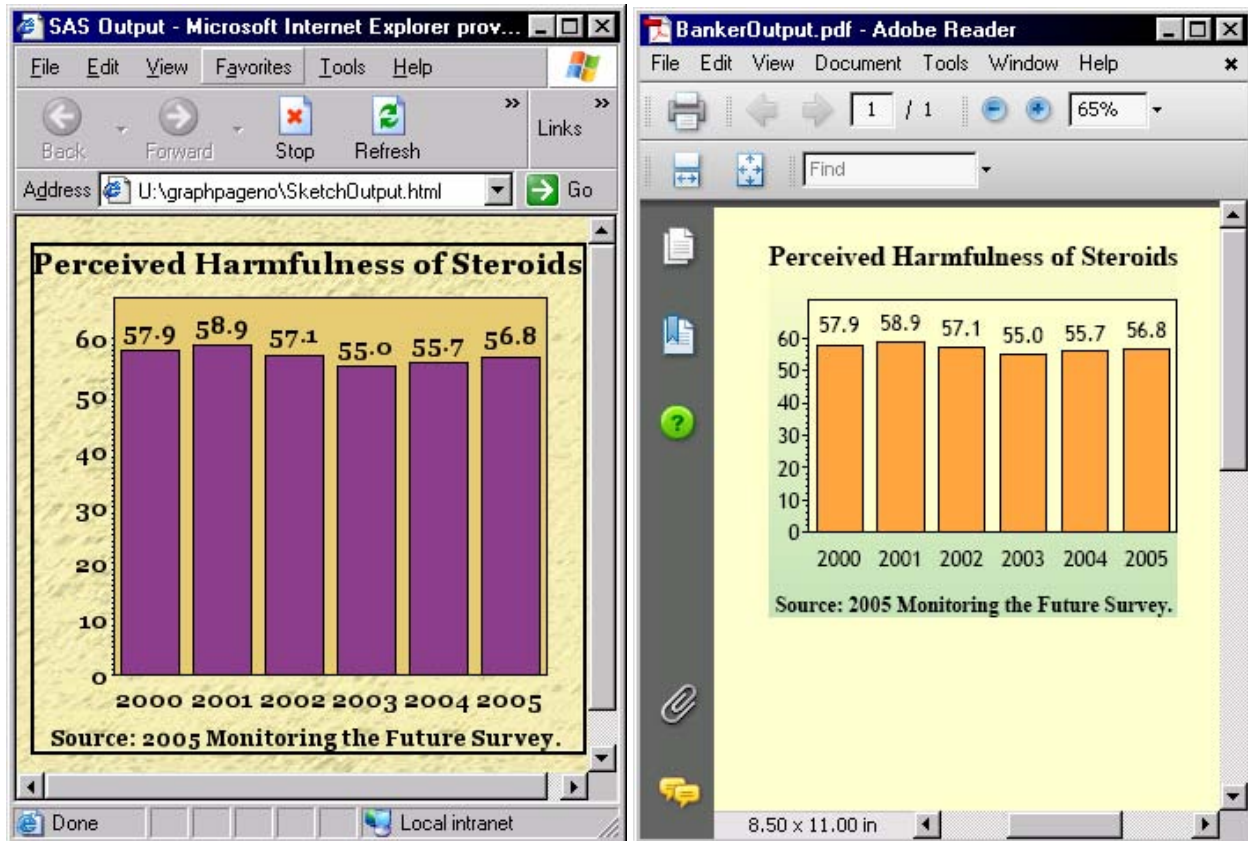


ODS DESTINATION INTERACTIONS

We have established that each output destination now has a unique style definition that control colors, fonts, graphic size, and a variety of additional graphical attributes. We also have established that each output destination has a unique default device to produce the best quality output. In the past, you have been instructed to run only one output destination at a time when producing graphical output because of various undesirable side effects. That is no longer true! Let's explore this further with the following code that creates Listing, HTML, and PDF file simultaneously using separate styles.

Example 6:

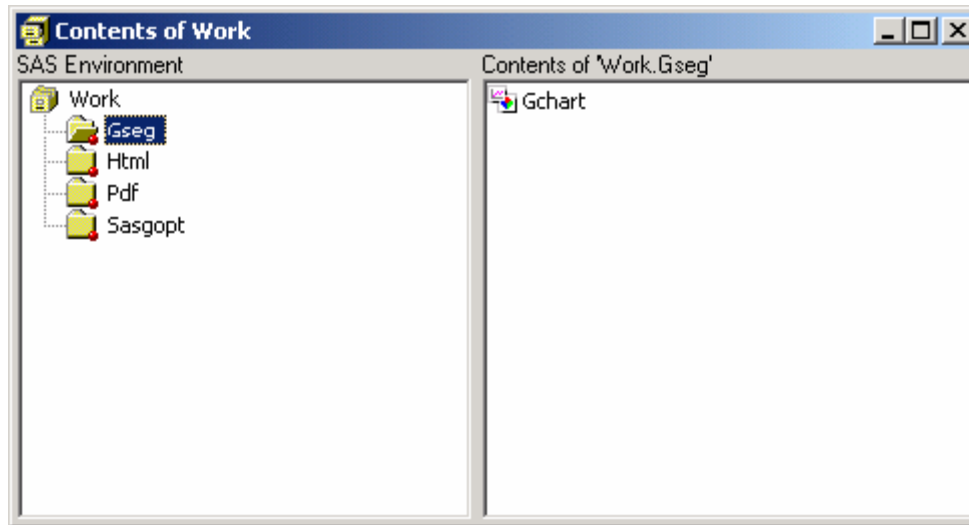
```
ods html file="SketchOutput.html" style=Sketch;
ods pdf "file="BankerOutput.pdf" style=Banker;
goptions hsize=3.25in vsize=3in border;
proc gchart data=sashelp.class;
vbar age;
run; quit;
ods pdf close;
ods html close;
```



Of course, the listing output will be saved in your WORK.GSEG catalog. If you ran this in DMS mode, your output would be displayed in the Graph Window as in previous releases.

CATALOG ENTRIES

Surely, all this new dynamic behavior is going to affect my catalog GSEG entries, and my GREPLAY code. Right? Well, the answer to that is both "Yes" and "No." Let's start with the "Yes" portion first. The graphical subsystem does in fact create a unique GRSEG entry for each graph in each output destination. However, when running with multiple output destinations, only the first active output destination saves GRSEG entries into the WORK.GSEG catalog ensuring that we retain backward compatibility with previous releases. In the default environment, SAS initialized the listing output destination first unless the customer has taken action to alter this behavior. So the "No" portion of the answer is that the last active device will still be used to create the GRSEG entry that is stored in the WORK.GSEG catalog just as in past releases. So where do all the other GRSEG entries go? When running with multiple output destinations SAS creates a new WORK catalog entry using the name of the output destination.



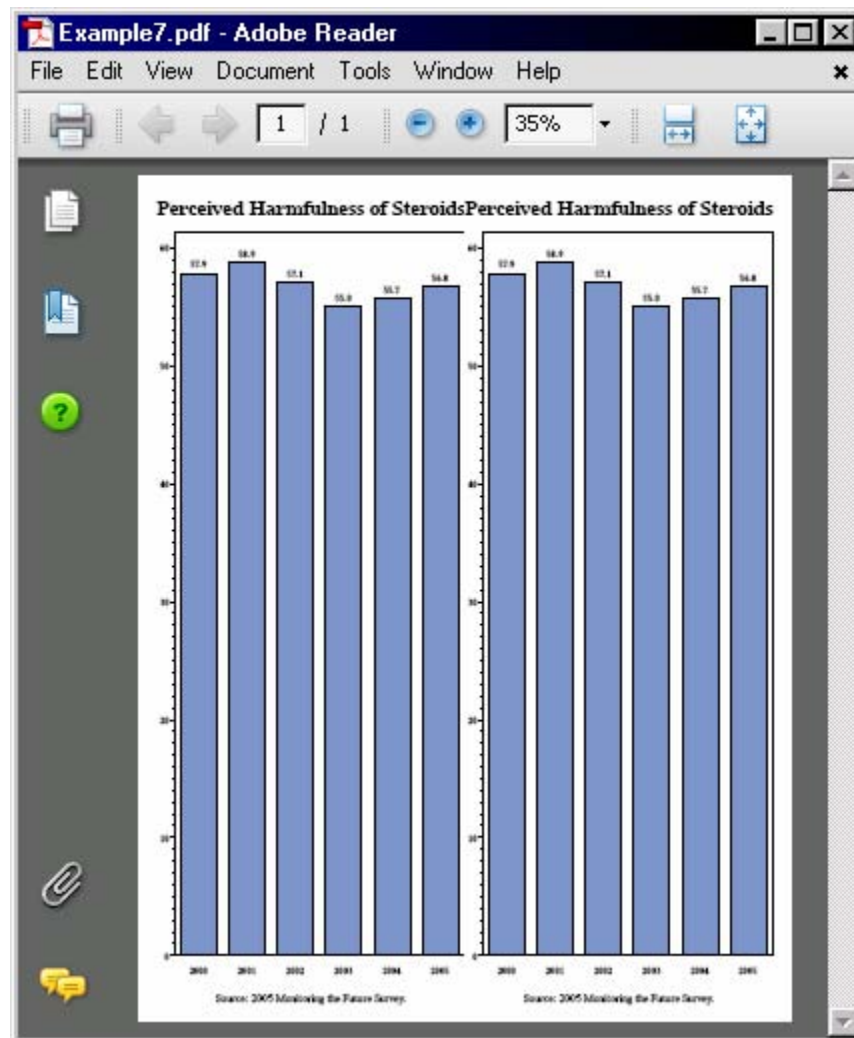
In this case, the following three entries will be created WORK.GSEG.GCHART.GRSEG, WORK.HTML.GCHART.GRSEG, and WORK.PDF.GCHART.GRSEG. Notice that we continue to use the same backward-compatibility naming convention except that we store them in unique catalogs. This allows the customer the ability to choose which catalog entry to replay when using your GREPLAY code.

DEFAULT SIZING

What determines the default size of my graph? We briefly touched on this earlier when we talked about graphic devices. Graphic devices contain a collection of default characteristics one of which is the default size. However, there are additional items that can affect the default size of a graph such as the following SAS global options: orientation, papersize, and PRINTERPATH, as well as ODS Layout, and the columns destination options. These options mostly affect the physical page-type output destinations like RTF and any of the PRINTER (PDF, PCL, PostScript) destinations.

Example 7:

```
ods listing close;
options orientation=landscape;
ods pdf file="Example7.pdf" columns=2;
title "Perceived Harmfulness of Steroids";
footnote "Source: 2005 Monitoring the Future Survey.";
axis1 label=none;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
    discrete maxis=axis1 raxis=axis1;
;run;quit;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
    discrete maxis=axis1 raxis=axis1;
;run;quit;
ods pdf close;
```



SAS will intuitively restrict the graphical output not to exceed the maximum column size, and will automatically retain the graph's default-device aspect ratio when altering the size.

CHANGING SIZE

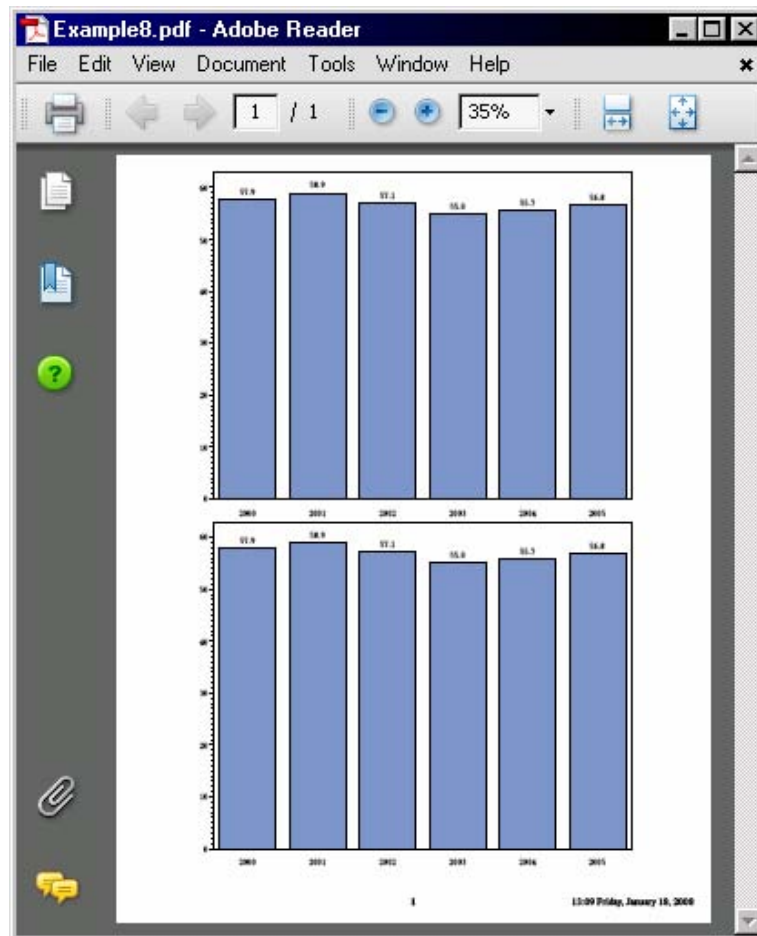
The recommended way to alter the default size of graphic output is by using the HSIZE and VSIZE graphic options just as in previous releases. However, SAS will continue to restrict the graphic output to the physically displayable area. For example, if you were to add “goptions hsize=8in vsize=12in;” to Example 8, the output would still be restricted to the maximum columns/page size. If either the HSIZE or the VSIZE graphic options have been specified independently, the other will automatically default to the device size. However if both have been specified, both will be honored barring any physical displayable limitations.

ODS PRINTER STARTPAGE SUPPORT

The STARTPAGE ODS statement option controls when page breaks occur. Normally page breaks occur on PROC boundaries, and when content fills the page. In previous releases, graphics output did not work well with the STARTPAGE option. Traditionally, all graph output was displayed on its own page and did not have a page number and date-time stamp support. All of these limitations have been addressed in SAS 9.2. I can produce multiple graphs on the same page by simply altering the size (goptions vsize=5in;), and by modifying the STARTPAGE option. By specifying “startpage=no”, a new page begins only when a page is filled. In the following example, I have also created a new style that modifies the default location of the date-time stamp, as well as the page number.

Example 8:

```
proc template;
define style Styles.Pageno;
parent=Styles.Printer;
class BodyDate /
  vjust=Bottom
  just=Right;
class Pageno /
  vjust=Bottom
  just=Center;
end; run;
ods pdf file="Example8.pdf" startpage=no style=Pageno;
options vsize=5in;
axis1 label=none;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
  discrete maxis=axis1 raxis=axis1;
;run;quit;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
  discrete maxis=axis1 raxis=axis1;
;run;quit;
ods pdf close;
```



MANAGING YOUR OUTPUT ON A PAGE

Today's reporting needs have evolved past the simple generation of reports and graphs and have matured into highly sophisticated executive summaries, elaborate forms, and brochure-quality presentations. These complex reporting requirements necessitate flexible page management control. ODS Layout continues to be the recommended way to address such reporting needs and remains in a preproduction state despite several enhancements in the graphics area. ODS Layout has two distinct varieties: ABSOLUTE and GRIDDED. Absolute layout allows you to specify the exact physical location for the placement of an item, while the more flexible gridded layout allows you to specify region (containers) that dynamically manage the content. Each has its benefits although absolute tends to be much more labor-intensive when constructing a sophisticated report. Absolute layout also is supported only in the ODS Printer destinations, while gridded layout is supported in HTML as well. ODS Layout follows the traditional ODS paradigm where statements wrapped (sandwich) your PROC code with a definitive starting and ending location. You can think of a gridded layout in a similar respect as a piece of graph paper or a spreadsheet with each cell being a displayable area for your procedure output. Here is some very simple syntax to provide a basis for our discussion.

Syntax:

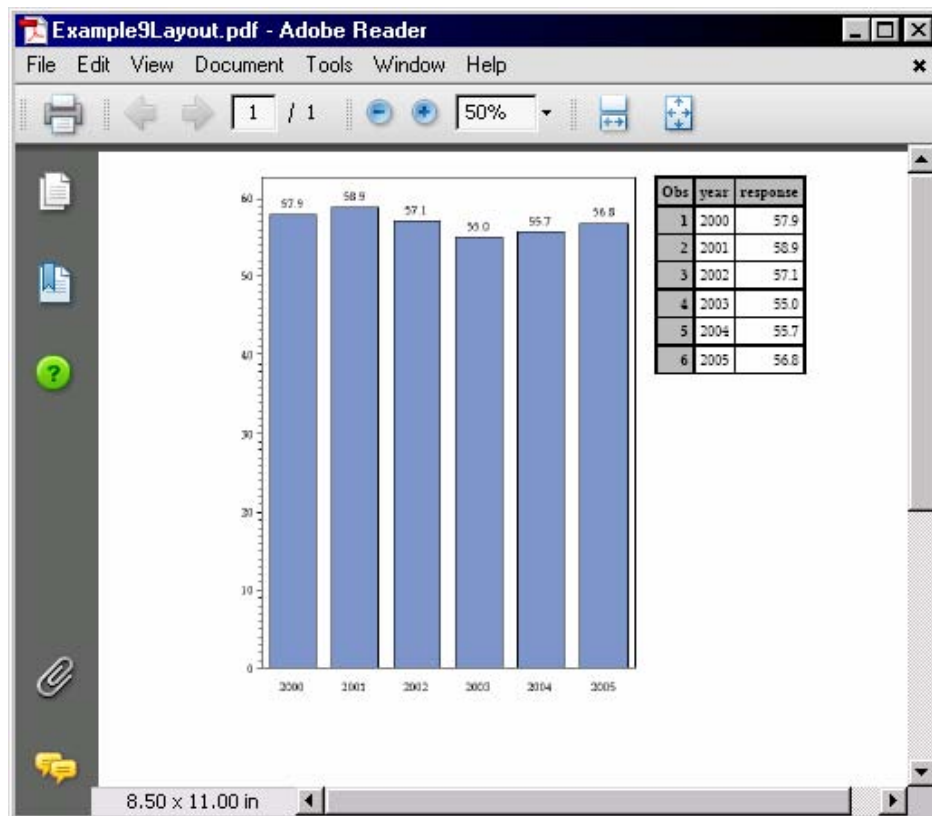
```
ODS LAYOUT START
  < Rows = < n >
    Columns = < n >
    Width = < dimension >
    Height = < dimension >
    Row_span = < n >
    Column_span = < n > > ;
ODS REGION
  < Width = < dimension >
    Height = < dimension >
    Row_span = < n >
    Column_span = < n > > ;
ODS LAYOUT END;
```

Example 9:

```
data harmful;
input year 4. @6 response 4.1;
output;
datalines;
2000 57.9
2001 58.9
2002 57.1
2003 55.0
2004 55.7
2005 56.8
;
ods listing close;
ods pdf file="Example9Layout.pdf";
ods layout start columns=2;

ods region width=4in;
axis1 label=none;
proc gchart;
vbar year / width=12 outside=sum sumvar=response
discrete maxis=axis1 raxis=axis1;
;run;quit;

ods region;
proc print;run;
ods layout end;
ods pdf close;
```



Notice that the ODS Layout specified a two-column format, and the size of the first column is specified on the ODS REGION statement. The ODS REGION statement indicates that we should move to the next available cell in the gridded (two-column) layout. Gridded regions are arranged in a column-wise fashion. Also, notice that the graph automatically is sized to the given region seeing that no GOPTIONS is specified.

ODS STATISTICAL GRAPHICS FRAMEWORK

ODS Graphics is a new production feature in SAS/GRAPH for creating statistical graphics. Many SAS products leverage this new functionality to automatically produce graphs such as SAS/STAT, SAS/ETS, SAS[®] High-Performance Forecasting, SAS/QC, and Base SAS. ODS Graphics Framework consists of three unique categories of technology: Graph Template Language (GTL), SAS/GRAPH Statistical procedures, and the ODS Graphics Editor.

AUTOMATIC GRAPHIC CREATION

Statistical graphs have unique requirements that differ from traditional business graphics. Often, there is additional statistical content which is valuable when interpreting the analysis or displaying a variety of statistics simultaneously for a comparative analysis maybe imperative to understand relationships in the data. When performing a statistical analysis in SAS, you can request that your results be displayed graphically as well as in the traditional tabular form by adding the "ods graphics on;" statement to your existing program. ODS Graphics is fully integrated into the ODS architecture and supports ODS styles as well as all of the commonly used output destinations.

GRAPHICS TEMPLATE LANGUAGE

One of the most powerful features that ODS provides is the ability to customize the procedure output that SAS produces. This is made possible because SAS ships the source code called the Table Template Language that describes the visual characteristics of each table of output produced. Therefore, customers have the unique ability to alter the SAS-supplied code to meet their particular needs. In that same vein, SAS supplies all the source code—called the Graph Template Language (GTL)—for graphs produced using the ODS Graphics Framework. SAS is shipping over 700 unique GTL graph definitions in SAS 9.2. These two languages share common concepts where appropriate, but also give the customer extensive customization capabilities to enhance their presentation effectiveness.

NEW PROCEDURES

SAS/GRAPH provides four new procedures: SGPLOT, SGSCATTER, SGPANEL, and SGRENDER. They provide a very simple syntax for creating graphs for commonly requested statistical displays. SGPLOT, SGSCATTER, and SGPANEL actually generate GTL code for you and then drive the ODS Statistical Framework to produce the graphical output. You will find these procedures invaluable when doing an initial statistical analysis. SGRENDER allows you to use existing GTL definitions with you new data.

GRAPHICS EDITOR

The ODS Graphics Editor is an interactive GUI-based editor that allows you to edit and annotate graphs that are produced using the ODS Graphics Framework. It allows you to add/edit titles, footnotes, axis labels, or add textual or graphic annotations; however, it does not allow you to alter the data integrity of the graphical display.

CONTACT INFORMATION

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

Daniel O'Connor
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
Work Phone: 1-919-531-6171
Fax: 1-919-677-4444
E-mail: Dan.OConnor@sas.com
Web: www.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.