

Paper RI-04

Converting SAS/GRAPH® to ODS Graphics
Jim Horne, Lowe's Companies, Inc., Mooresville, NC**ABSTRACT**

As of SAS 9.3, SAS has moved ODS Graphics and the Statistical Graphics procedures from SAS/GRAPH® to Base SAS®. This provides us an opportunity to eliminate SAS/GRAPH by converting our SAS/GRAPH procedures to ODS Graphics procedures.

This paper is an overview of some ways to convert basic graphs from SAS/GRAPH to ODS Graphics. It provides a brief introduction to ODS Graphics and the SAS statements required to run the ODS Graphics procedures. It then compares some basic SAS/GRAPH and ODS Graphics output along with the statements required to produce each graph so you can see the similarities and the differences. It does not provide an exhaustive introduction to ODS Graphics or an introduction to graphing using SAS software. Familiarity with SAS/GRAPH is assumed.

INTRODUCTION

We will first mention the SAS/GRAPH statements that we know are going away. Then we will look at statements still used, both ones that we must have in order to execute and one that we may use but do not have to use. Finally in this section, we will look at the ODS GRAPHICS statement, which is required in order to generate ODS Graphics.

We will then briefly go through an overview of the ODS Graphics procedures and mention the Graph Template Language (GTL). We will not spend much time on it because this course is not an explanation of those procedures but rather an introduction to them, with examples of how to use them.

We then go into the heart of the presentation, the comparison of how to generate graphs in SAS/GRAPH as opposed to ODS Graphics. This is where the bulk of our time will be spent. Our last major topic is actually a small hodgepodge of things, as we look at doing things that were either not possible in SAS/GRAPH, or too tedious to do. Finally, we wrap things up by reviewing everything we covered.

SAS STATEMENTS

All of the SAS/GRAPH statements go away with ODS Graphics. These include the LEGEND, SYMBOL, AXIS and GOPTIONS statements as well as all of the SAS/GRAPH procedures. The FILE and ODS HTML statements are still required to generate ODS output and are used the same way we use them when we produce SAS/GRAPH output. The TITLE, FOOTNOTE, NOTE, FORMAT, LABEL BY and WHERE statements can all be used with ODS Graphics procedures. The TITLE, FOOTNOTE and NOTE statements can be used to put titles, footnotes and notes into your graphs just as they can be today. If you want to change the LABEL or FORMAT of variables just for the procedure call, you can do that as well, just as you could in SAS/GRAPH. Using BY and WHERE to group and subset your data is also allowed, just as it was in SAS/GRAPH.

A new statement that is required is the ODS Graphics statement. We will not provide a full description of it here but will cover some options. A complete description can be found in the SAS 9.3 Output Delivery System User's Guide. It can be found at <http://support.sas.com/documentation/cdl/en/odsug/62755/HTML/default/viewer.htm#titlepage.htm>.

The simplest form is either ODS GRAPHICS ON or ODS GRAPHICS OFF. ON enables graphs to be produced while OFF disables them. Other options include to IMAGENAME, IMAGEMAP, HEIGHT, WIDTH and RESET. IMAGENAME allows you to name each graph (a practice the author strongly encourages). IMAGEMAP enables flyover data tips and drill down functionality. HEIGHT and WIDTH can be used to control the size of graphs. RESET allows any number of options to be reset to their default values. Some examples of this statement are

- ODS GRAPHICS ON;
- ODS GRAPHICS OFF;
- ODS GRAPHICS ON / IMAGENAME=xxxx ;
- ODS GRAPHICS ON / IMAGENAME=xxxx IMAGEMAP ;
- ODS GRAPHICS ON / RESET=ALL ;

ODS GRAPHICS PROCEDURES

The ODS Graphics procedures are SGDESIGN, SGPANEL, SGPLOT, SGRENDER and SGSCATTER.

The SGDESIGN procedure processes files created by the ODS Graphics Designer application and will not be discussed here.

The SGPANEL procedure produces a set of graphs in a panel format for one or more classification variables. There is nothing quite like it in SAS/GRAPH that this author is aware of. The closest thing to it is PROC GREPLAY with its ability to put graphs from the same or different data sets together on a single web page. While PROC GREPLAY in SAS/GRAPH offers more flexibility in terms of designing a single page with multiple graphs, SGPANEL offers ease of use when the same dataset can be used to graph different things that you want to show together. An example of it is shown in the EXTRAS section

The SGSCATTER procedure produces a paneled set of graphs for multiple combinations of variables. It has no equivalent function that the author is aware of in SAS/GRAPH but will not be covered further here. If you only want to produce a single scatter plot as opposed to a combination of them, it can be done using PROC SGPLOT.

The SGPLOT procedure is the workhorse of the ODS Graphics procedures, combining the capabilities of both GCHART and GPLOT from SAS/GRAPH. Below are some but not all of the graphs it can generate. For full details, see the SGPLOT section of ODS Graphics: Procedures Guide at <http://support.sas.com/documentation/cdl/en/grstatproc/64778/HTML/default/viewer.htm#titlepage.htm>.

- | | |
|---------------|-------------|
| • BAND | • NEEDLE |
| • BUBBLE | • REFLINE |
| • DENSITY | • REG |
| • DOT | • SCATTER |
| • HBAR/VBAR | • SERIES |
| • HBOX/VBOX | • STEP |
| • HIGHLOW | • VECTOR |
| • HISTOGRAM | • WATERFALL |
| • HLINE/VLINE | |

Table 1. Selected SGPLOT Plot Types

The SGRENDER procedure produces graphical output from templates that are created with the Graph Template Language (GTL). GTL is a comprehensive language for creating statistical graphics. It can be used to create customized layouts and graphs that are beyond the scope of the ODS Graphics procedures. For more information about the GTL, see the SAS Graph Template Language: Reference at <http://support.sas.com/documentation/cdl/en/grstatgraph/64764/HTML/default/viewer.htm#titlepage.htm> and the SAS Graph Template Language: User's Guide at <http://support.sas.com/documentation/cdl/en/grstatug/63302/HTML/default/viewer.htm#titlepage.htm>. PROC TEMPLATE creates the form of the custom graph and PROC SCGRENDER then applies it against a specific dataset. We will see a quick example of this when we look at pie graphs.

GRAPH COMPARISONS

Now we are ready to compare creating graphs in SAS/GRAPH and by using ODS Graphics. We will look at side by side comparisons of the code and of the graphs it produces. We will use a minimum of special statements to generate these graphs. They are as follows:

Web page definition

```
ODS LISTING CLOSE;
FILENAME odsout "/a-file-definition";
ODS HTML BODY="odsclass.html" (TITLE="ODS Class")
  STYLE=htmlblue
  PATH=odsout (URL=NONE) ;
```

SAS/GRAPH statement

```
SYMBOL1 INTERPOL=JOIN;
```

ODS Graphics statement

```
ODS GRAPHICS ON / IMAGENAME='odsclass'
NOBORDER NOANTIALIAS HEIGHT=725px ;
```

The web page definition statements consist of an ODS LISTING CLOSE statements so that the LISTING destination will not get errors trying to produce graphics output; a FILENAME statement to define an output directory or PDSE, and an ODS HTML statement to define the body of our graphs. If you use different FILENAME statements or more complicated ODS HTML statements, they should continue to work as they did before. Having said that, all the author can state for certain is that his statements needed no modification to work. We use the page style HTMLBLUE because that is the SAS 9.3 default.

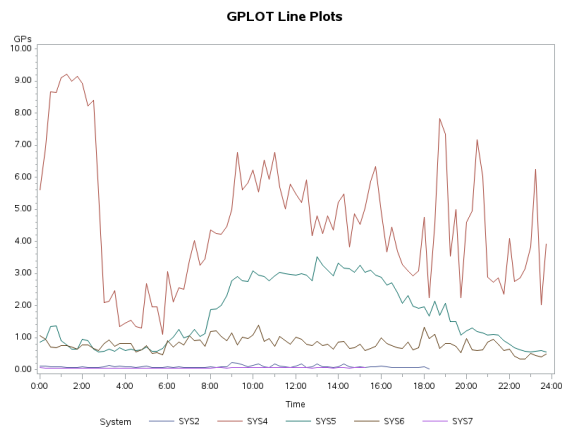
For the SAS/GRAPH graphs produced, the only statement added beyond the defaults was a SYMBOL statement to connect the points.

The ODS GRAPHICS statements use turned on ODS Graphics, names the output it produces, turns off antialiasing and borders and sets the height of the graphs produced to 725 pixels.

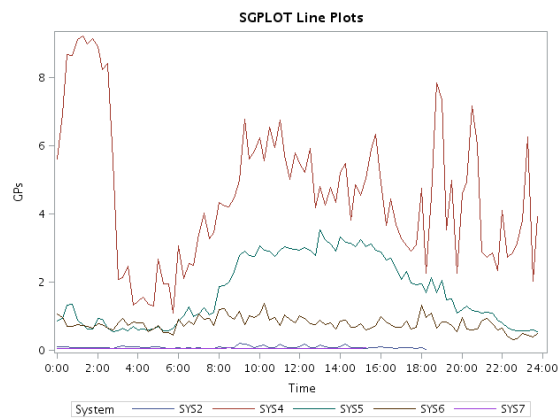
We will now go through each graph type, starting with the SAS statements they use, proceeding to the graphs themselves and concluding each section with any additional explanation needed.

LINE PLOTS

SAS/GRAPH
 PROC GPLOT DATA=cpr70lp ;
 TITLE "GPLOT Line Plots" ;
 PLOT usedgps*time=system ;
 RUN;



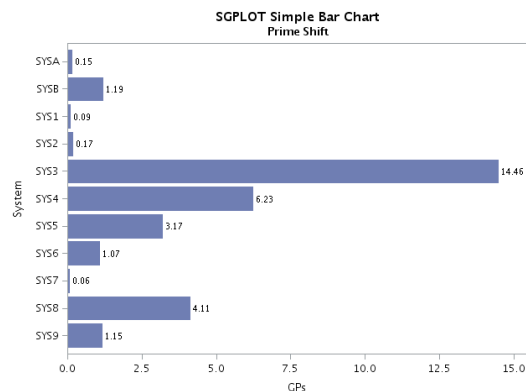
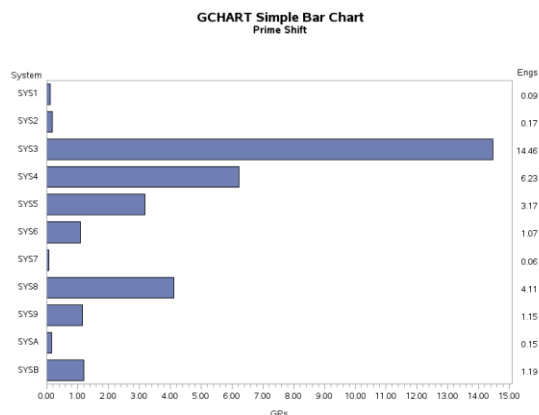
ODS Graphics
 PROC SGPLOT DATA=cpr70lp ;
 TITLE "SGPLOT Line Plots" ;
 SERIES X=time Y=usedgps / GROUP=system ;
 RUN;



SIMPLE BAR CHARTS

SAS/GRAPH
 PROC GCHART DATA=meanengs ;
 BY date DESCENDING shift ;
 TITLE "GCHART Simple Bar Chart" ;
 TITLE2 "#BYVAL(shift) Shift" ;
 HBAR system / DISCRETE SUMVAR=usedgps
 SUM
 SUMLABEL = 'Engs' ;
 RUN;

ODS Graphics
 PROC SGPLOT DATA=meanengs CYCLEATTRS ;
 BY date DESCENDING shift ;
 TITLE "SGPLOT Simple Bar Chart" ;
 TITLE2 "#BYVAL(shift) Shift" ;
 HBAR system / DATALABEL
 FILL
 RESPONSE=usedgps
 STAT=sum ;
 RUN;

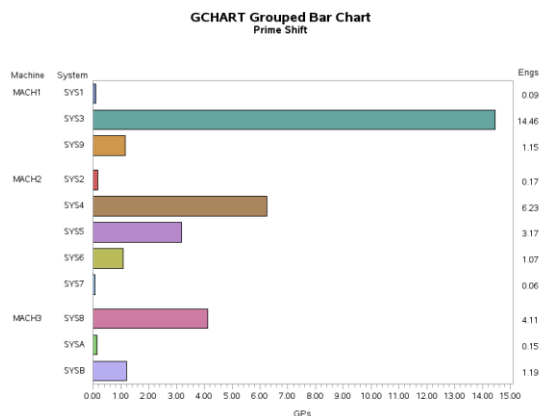


Option CYCLEATTRS was used on the PROC SGPLOT statement to force a different color than would have been produced without it. It did not force multiple colors; it only made a different choice of the one that was used.

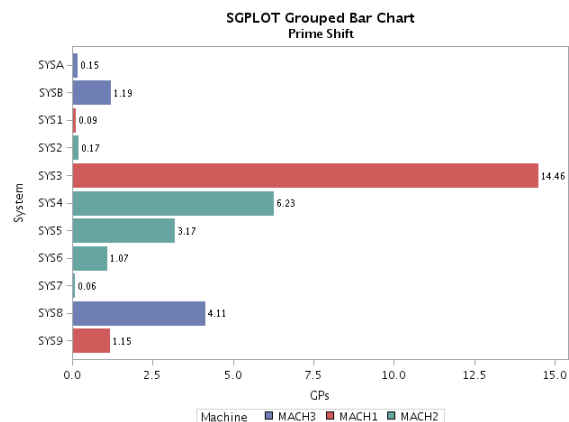
Where PROC GCHART used a SUMLABEL to give a title to the bar values that were written out with the SUM statement, PROC SGPLOT uses the DATALABEL option to produce values which appear immediately adjacent to the end of each bar. Additionally, the FILL option is required to produce bars that have colored insides and the STAT option is required to say what operation to perform on the data.

GROUPED BAR CHART

```
SAS/GRAPH
PROC GCHART DATA = meanengs ;
BY date DESCENDING shift ;
TITLE "GCHART Grouped Bar Chart" ;
TITLE2 "#BYVAL(shift) Shift" ;
HBAR system / DISCRETE SUMVAR=usedgps
SUM
SUMLABEL = 'Engs'
GROUP = machine
PATTERNID = MIDPOINT
HTML = gpdrill
NOZERO ;
RUN;
*gpdrill= CATS('HREF="gpplt',machnum,'.html'');
```



```
ODS Graphics
PROC SGPLOT DATA=meanengs CYCLEATTRS ;
BY date DESCENDING shift ;
TITLE "SGPLOT Grouped Bar Chart" ;
TITLE2 "#BYVAL(shift) Shift" ;
HBAR system /
DATALABEL
FILL
GROUP = machine
RESPONSE = usedgps
STAT = sum
URL = urlgp ;
RUN;
*urlgp = CATS('gpplt',machnum,'.html');
```



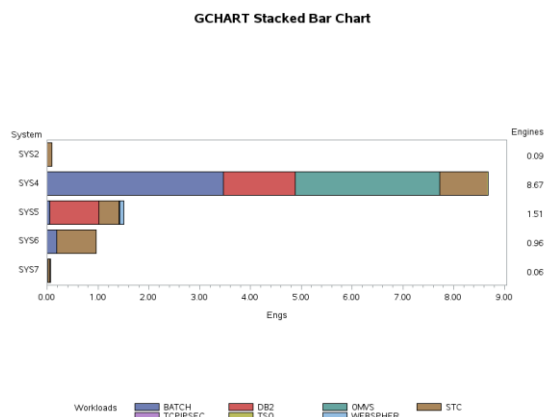
The biggest difference in the grouped bar charts is the effect of grouping. In SAS/GRAPH, the data is actually grouped according to the GROUP specified. In ODS Graphics the data order is unchanged and only color indicates the group to which an object belongs. In the SAS/GRAPH output, color still has no significance.

This may not look like a good way to group data in ODS Graphics but we will see later that PROC SGPanel offers us equivalent function to SAS/GRAPH.

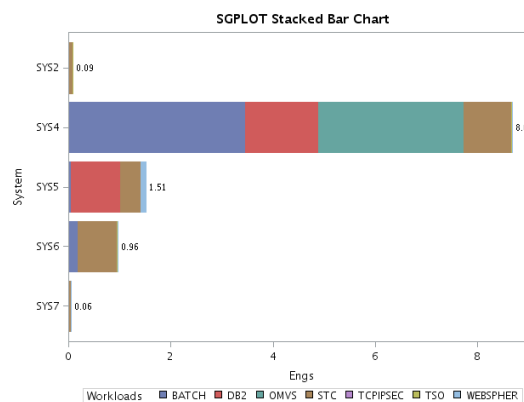
This example also shows the different way drill down capability is indicated in SAS/GRAPH and ODS Graphics. The HTML= option of SAS/GRAPH has an equivalent in the URL= option of ODS Graphics. But SAS/GRAPH required a link to be in the form HREF=link, surrounded by quotes. ODS Graphics simply requires the name of the link.

STACKED BAR CHARTS

```
SAS/GRAPH
PROC GCHART DATA=freq72go ;
BY date DESCENDING shift ;
TITLE "GCHART Stacked Bar Chart" ;
HBAR system / DISCRETE SUMVAR=gpwkld
SUM
SURLABEL = "Engines"
NOZERO
SUBGROUP = wkldname
PATTERNID = SUBGROUP
HTML = ghtml
HTML_LEGEND = bardrill ;
RUN;
```



```
ODS Graphics
PROC SGPlot DATA=freq72go ;
BY date DESCENDING shift ;
TITLE "SGPlot Stacked Bar Chart" ;
HBAR system / RESPONSE=gpwkld
DATALABEL
FILL
GROUP = wkldname
STAT = sum
URL = urlwkchtgp ;
RUN;
```



The NOZERO option is required in SAS/GRAPH to prevent systems which were in other groups but not the displayed one from showing up. This is not necessary in ODS Graphics.

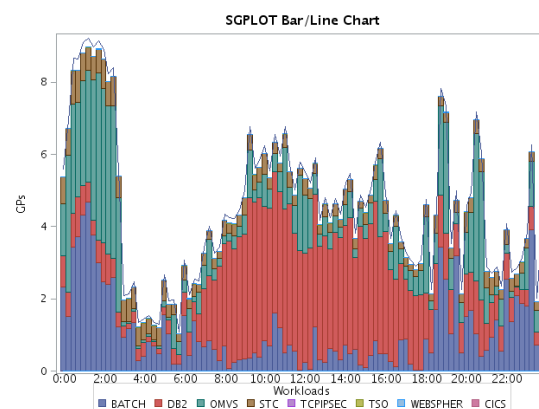
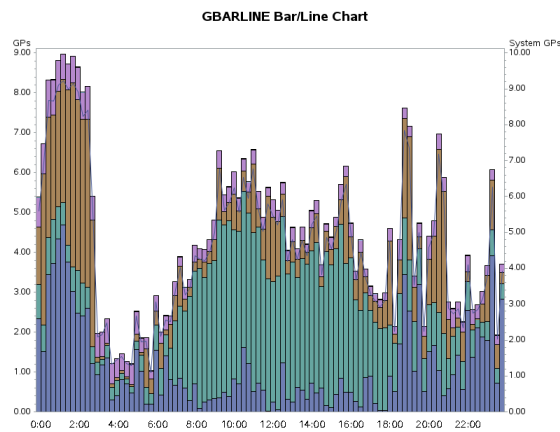
The ability to drill down to another web page from the SAS/GRAPH legend is an option that is not available in ODS Graphics. You can only drill down from the data, not from its legend.

Also note that SAS/GRAPH shrinks the graph to keep the bars the same size. ODS Graphics does not do this by default.

COMBINED BAR/LINE CHARTS

```
SAS/GRAPH
AXIS99 LABEL=NONE VALUE=NONE NOPLANE ;
PROC GBARLINE DATA=summwkld ;
  TITLE "GBARLINE Bar/Line Chart" ;
  BAR time / SUMVAR=scgps
    SUBGROUP = wkldname
    ANNOTATE = annowkld
    MAXIS = axis99
    DISCRETE
    SPACE = 0 ;
  PLOT / SUMVAR=usedgps ;
RUN;
```

```
ODS Graphics
PROC SGPLOT DATA=summwkld
  SGANNO=annotime
  TITLE "SGPLOT Bar/Line Chart" ;
  VBAR time / RESPONSE=scgps
    GROUP=wkldname
    BARWIDTH=1 ;
  VLINE time / RESPONSE=usedgps
    GROUP=wkldname ;
  XAXIS DISPLAY=(NOLABEL NOTICKS
  NOVALUES) ;
RUN;
```



In PROC GBARLINE output no LEGEND is shown by default; it must be requested. In the SGPLOT, you do get a legend by default, but you must use the GROUP option on all requests if you use them on any. This can be a little awkward, such as in this example. The desired VLINE plot is for used GPs in the entire CEC but the only way to indicate that is to create your own legend using annotation. As an alternative you can create your own graph template and render it but that is beyond the scope of this paper. Also see the SPACE=0 option in GBARLINE and the equivalent BARWIDTH=1 option in SGPLOT. These options ensure there is no white space between bars unless you want it.

Note also that we have used an extra SAS/GRAPH statement, AXIS99 and an SGPLOT option, XAXIS. The purpose of both of these statements is to make sure no values are printed on the X axis by default as we want to annotate an axis on each. In GBARLINE this is accomplished using the ANNOTATE parameter of the BAR statement. In SGPLOT, it is accomplished using the SGANNO parameter on PROC SGPLOT invocation.

Note in this example that the default legend is not in alphabetic order, but is in the order of the data. Since there is no CICS workload data in the sample we have (it is present in data we do not see) the workload for CICS is at the end of the legend. This can be changed, but it is important to be aware of the default behavior.

ODS Graphics also provides an ability to create an attribute map and always have the same workload have the same color, even if the workloads change over time. This is something that will not be discussed here because of time limitations, but it is something new that did not exist in SAS/GRAPH.

TIME AXIS ANNOTATION

```
SAS/GRAPH
DATA annowkld ;
SET summwkld (KEEP=date machine system time) ;
IF MOD(time,7200) = 0 THEN DO;
  function = 'label' ;
  color    = 'black' ;
  position = '8' ;
  when     = 'A' ;
  STYLE    = 'simplex';
  hsys     = '5' ;
  xsys     = '2' ;
  ysys     = '1' ;
  size     = 2.5 ;
  x        = time ;
  y        = 0 ;
  text = PUT(time,TIME5.) ;
END;
RUN;
```

```
ODS Graphics
DATA annotime ;
function = 'TEXT' ;
justify  = 'RIGHT' ;
x1space  = 'DATAVALUE' ;
y1space  = 'WALLPERCENT' ;
y1       = -1.5 ;
DO x1 = '00:00'T TO '22:00'T BY '02:00'T ;
  label = PUT(x1,TIME5.) ;
OUTPUT;
END;
RUN;
```

As you may be aware, bar graphs produce an axis entry for each bar by default. In SAS/GRAPH the only way to change this behavior was to blank out axis values and annotate your own. ODS Graphics offers a way to skip every other observation within the procedure itself, but annotation offers the best way to trim the axis values down to look their best without losing any of the individual bars.

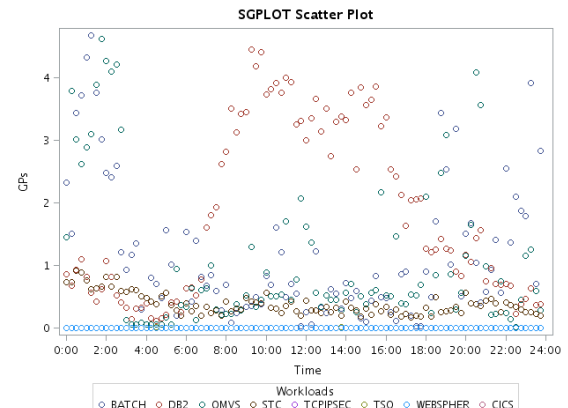
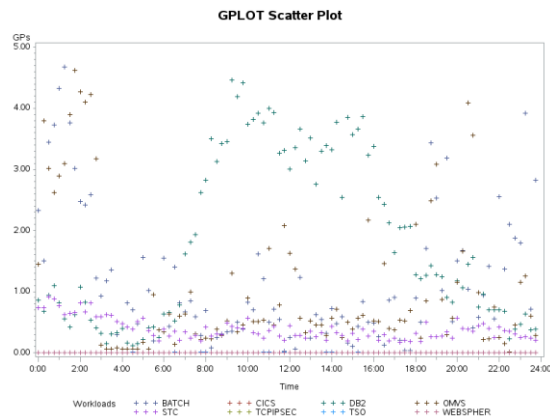
And the annotation datasets are created differently for SAS/GRAPH and ODS Graphics. In SAS/GRAPH, there must be an observation in the annotate dataset for every observation in the base dataset. This means that you need separate annotate datasets for workload graphs than you do service class graphs, because the base datasets themselves are different.

In ODS Graphics, you only need observations in the annotate dataset of things that actually have something to write. In this example, of an X axis with time values being written every two hours, we only need thirteen observations. Additionally, we can use this dataset wherever we need this annotation. We do not have to create a new one for every base dataset.

SCATTER PLOTS

```
SAS/GRAPH
symbol1;
PROC GPLOT DATA=summwkld ;
  BY date machine system ;
  TITLE "GPLOT Scatter Plot" ;
  PLOT scgps*time=wkldname ;
RUN;
```

```
ODS Graphics
PROC SGPLOT DATA=summwkld ;
  BY date machine system ;
  TITLE "SGPLOT Scatter Plot" ;
  SCATTER X=time Y=scgps /
    GROUP=wkldname ;
RUN;
```

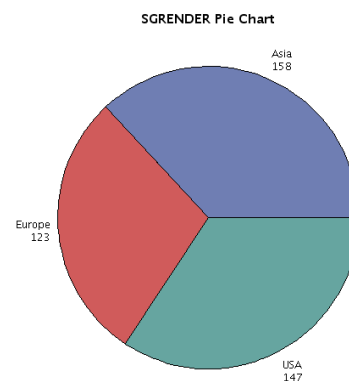
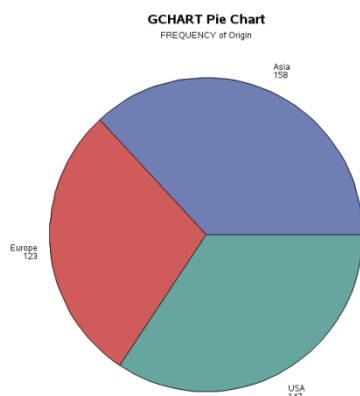


In comparing one plot scatter plots between SAS/GRAPH and ODS Graphics, we first need to determine what a scatter plot is – namely, a collection of points plotted on the same set of axes. In ODS Graphics we are given the SCATTER statement in PROC SGPLOT to do this. In SAS/GRAPH we must define a symbol with no interpolation to it and then use PLOT to lay down our data. In fact, this is actually the default SYMBOL definition, but it is not a default you see taken often. With that, the scatter plots are remarkable similar. Both go through colors and then symbol shapes to define all the groups they need.

PIE CHARTS

SAS/GRAPH
 PROC GCHART DATA=sashelp.cars ;
 TITLE 'GCHART Pie Chart' ;
 PIE origin /
 FILL=solid
 SLICE=outside
 VALUE=outside
 COUTLINE=BLACK ;
 RUN;

ODS Graphics
 PROC TEMPLATE;
 DEFINE STATGRAPH simplepie;
 BEGINGRAPH;
 ENTRYTITLE 'SGRENDER Pie Chart' ;
 LAYOUT REGION;
 PIECHART CATEGORY=ORIGIN /
 DATA LABELLOCATION=OUTSIDE;
 ENDLAYOUT;
 ENDGRAPH;
 END;
 RUN;
 PROC SGRENDER DATA=sashelp.cars
 TEMPLATE=simplepie;
 RUN;



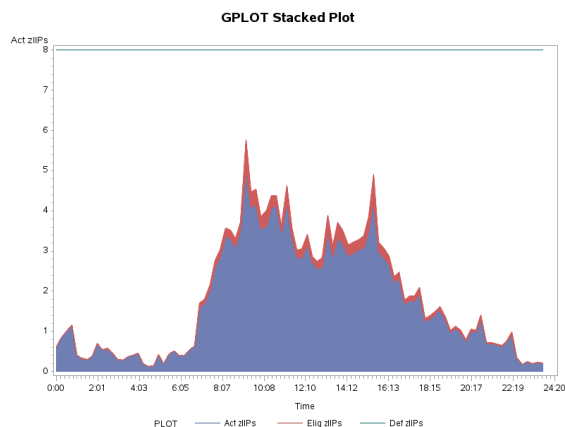
Pie charts are not very common and that may be why ODS Graphics, as of SAS 9.3, only makes it available through the Graph Template Language (GTL) and not through a PIE statement in PROC SGPLOT.

This example is taken straight from SAS documentation and you can see that both sets of procedures produce the same graph.

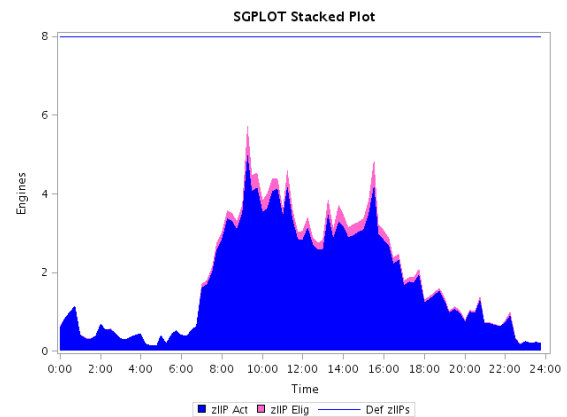
This is one of the simplest of graph templates. Every statement in the template you see here is required. There are no optional extras – but there could be. For more information see the SAS 9.3 ODS Graphics GTL Reference and GTL User's Guide.

STACKED PLOTS

```
SAS/GRAPH
PROC GPLOT DATA=cpr70lp ;
  BY date machine system ;
  FORMAT sczips sczipa lparzips 3. ;
  TITLE "GPLOT Stacked Plot" ;
  PLOT (sczips sczipa lparzips)*time / OVERLAY
    AREAS = 2
    LEGEND
    HTML_LEGEND = combrpt
    HTML      = combrpt ;
RUN;
```



```
ODS Graphics
PROC SGPLOT DATA=cpr70lp ;
  TITLE "SGPLOT Stacked Plot" ;
  BY date machine system ;
  BAND X=time LOWER=0 UPPER=sczips /
    FILL FILLATTRS=(COLOR=CX0000FF)
  LEGENDLABEL='zIIP Act' ;
  BAND X=time LOWER=sczips UPPER=sczipa /
    FILL FILLATTRS=(COLOR=CXFF66CC)
  LEGENDLABEL='zIIP Elig' ;
  SERIES X=time Y=lparzips /
    LINEATTRS=(PATTERN=1 COLOR=CX0000FF)
  URL=urlcombrpt ;
  YAXIS MINOR LABEL='Engines' INTEGER ;
RUN;
```



In this example of stacked plots we must prepare the data the same regardless of whether we use SAS/GRAPH or ODS Graphics. We must ourselves stack the data. In series or line plots, unlike bars, SAS does not perform automatic stacking for us. We have to do it ourselves.

Note that in SAS/GRAPH we control coloring areas of the graph with the AREAS option of the PLOT statement; in ODS Graphics we use BAND graphs with explicitly defined upper and lower bounds. If we want a single line we use a SERIES statement. The two BAND and one SERIES statement you see in the ODS Graphics example contrast to the PLOT OVERLAY statement in SAS/GRAPH.

Note too that while SAS/GRAPH supports drill down capability for all parts of the graph including the legend, ODS Graphics only allows the SERIES statement to enjoy drill down capability. It is not available in BAND statements or in the legend. We also show how you can color individual PLOT statements in ODS Graphics when it makes sense. Not that you must do it, but you can if you want to do it.

EXTRAS

Now we get to some extras, things that were not in SAS/GRAPH but are in ODS Graphics, or were so different in SAS/GRAPH that it is easier to start over as if they were brand new functions.

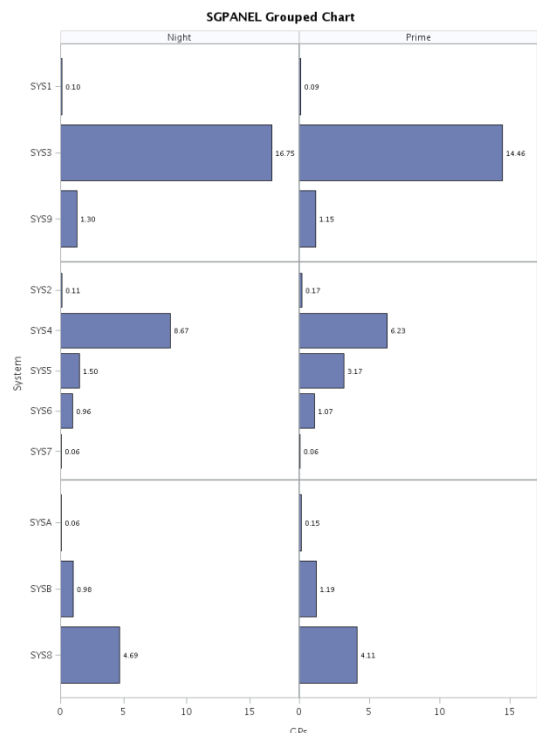
PANELS

Here is an example of SGPPANEL, a way to categorize data and produce multiple graphs in a single web page. The example shown is one we saw earlier in grouped data. Now we see that there were actually two shifts. With SAS/GRAPH each shift was shown on a separate web page. Using SGPPANEL, we can build one single paneled graph showing all data from both shifts.

```

ODS GRAPHICS ON / WIDTH=900px ;
PROC SGPanel DATA=meanengs CYCLEATTRS ;
BY date ;
TITLE "SGPanel Grouped Chart" ;
PANELBY shift machine / LAYOUT=lattice
UNISCALE=column
NOVARNAME ;
HBAR system / RESPONSE=usedgps
DATALABEL
URL=urlgp ;
COLAXIS OFFSETMIN=0;
RUN;

```



CUSTOM DATA TIPS

We have talked about PROC SGRENDER and the Graph Template Language before. Here is another example showing when you might want to build a template and render your own graph rather than just use PROC SGPLOT. The graphs these two sets of procedures produce look the same which is why no example is shown.

The difference between them comes if you want to use the flyover, or data tips, information. In SGPLOT, the data tips consist of the X variable and value, the Y variable and value and, if present, the GROUP variable and value. In our example, that gives us the service class period as a single variable, the sysplex and system this is for, again combined as a single variable and the group variable MAPPI and its value. In this case, MAPPI was a computed variable set to be 0, 1 2 or 3, depending on the value of the performance index, perfindx. As such it is a meaningless variable and value – perfindx has what we really want.

So we create the template on the right. This gives us service class and period as individual variables, sysplex and system as individual variables, and in place of the computed category variable MAPPI, we get the performance index. You may not use this very often but it is good to know that you can.

```

PROC SGPlot DATA= pimean ;
BY date ;
SCATTER X=svrcl Y=sysplexsys /
GROUP=mappi
URL=urlpi
MARKERATTRS=(SIZE=15px
SYMBOL=circlefilled) ;
XAXIS DISCRETEORDER=unformatted ;
RUN;

```

```

PROC TEMPLATE;
DEFINE STATGRAPH pisummplot ;
BEGINGRAPH;
ENTRYTITLE "SGRENDER PI SUMMARY for "
_BYVAL_ ;
LAYOUT overlay ;
SCATTERPLOT X=svrcl Y=sysplexsys /
GROUP=mappi
URL=urlpi
NAME='pisumm'
MARKERATTRS=(SIZE=15px
SYMBOL=circlefilled)
ROLENAMES=( _pi=perfindx _plex=sysplex
_sys=system _svcl=svrcl _per=period)
TIP=( _plex _sys _svcl _per _pi ) ;
DISCRETELEGEND 'pisumm' /

```

```
TITLE="PI Level" ;  
ENDLAYOUT;  
  
ENDGRAPH;  
END;  
RUN;  
PROC SGRENDER DATA=pimean  
TEMPLATE=piummplot ;  
BY date;  
RUN;
```

CONCLUSION

We know that SAS has moved ODS Graphics from SAS/GRAPH to Base SAS. We have gone through the statements that will go away and new statements that will be needed. We have taken a quick look at the new ODS Graphics procedures that can potentially replace the SAS/GRAPH procedures we already know. We have seen some simple examples that show how to create the same tasks using each method. You can now use this basic knowledge to evaluate whether a conversion from SAS/GRAPH to Base SAS is right for your graphing needs.

REFERENCES

- All SAS 9.3 manuals are available at <http://support.sas.com/documentation/onlinedoc/base/index.html>
- Under SAS Procedures Guides, see
 - SAS 9.3 Output Delivery System: User's Guide
 - SAS 9.3 ODS Graphics: Procedures Guide
- Under ODS Graphics, see
 - SAS 9.3 Graph Template Language: User's Guide
 - SAS 9.3 Graph Template Language: Reference

RECOMMENDED READING

- Statistical Graphics in SAS, Warren F. Kuhfeld, 2010, ISBN 978-1-60764-485-9
- Using PROC SGPLOT for Quick High Quality Graphics, Delwiche and Slaughter, <http://www.wuss.org/proceedings08/08WUSS%20Proceedings/papers/how/how05.pdf>
- Graphics Sample Output Gallery, http://support.sas.com/sassamples/graphgallery/PROC_SGPLOT.html
- SAS Notes and Concepts for ODS, <http://support.sas.com/rnd/base/ods/templateFAQ/index.html>

CONTACT INFORMATION

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

Name:	Jim Horne
Enterprise:	Lowe's Companies, Inc.
Address:	1000 Lowes Blvd
City, State ZIP:	Mooresville, NC 28117
Work Phone:	704-758-5354
E-mail:	jim.s.horne@lowes.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.