# Multiple Graphs on One page, the easy way (PDF) and the hard way (RTF)

Kevin P. Delaney MPH, Northrop-Grumman Mission Systems, Atlanta, GA

#### Abstract:

SAS/GRAPH® software can be used in combination with the Output Delivery System (ODS) to produce high-resolution graphics in production reports. It is fairly easy to use the ODS to insert one graph at a time into either a Portable Document Formatted (PDF) or Rich Text Formatted (RTF) report. However, trying to insert multiple graphs on one page can be a little more of a challenge. For ODS PDF, the option STARTPAGE=NEVER and some SAS/GRAPH options which control the size of the graphs can be used to place multiple graphs on one page. Unfortunately, STARTPAGE=NEVER does not work very well in the RTF destination, so a solution which utilizes PROC GREPLAY and DDE is offered. These techniques, including code and examples, are presented, and a potential solution using the new Version 9 ODS LAYOUT capabilities is also be discussed

## Introduction

SAS/GRAPH software is designed to produce high quality graphical representations of your data. Often a project will require output that presents only one graph per page. By using the Output Delivery System (ODS) it is possible (and relatively simple) to export your graphs to either a Portable Document Format (PDF) file or a Rich Text Formatted (RTF) file (for use in Microsoft Word). However, there are some reports for which it would be better to present multiple graphs on a single page.

Achieving this effect with the ODS can be a challenge. However, like almost every SAS solution, there are multiple ways that this goal can be accomplished. This paper presents solutions for both the PDF and the RTF destinations. The paper also provides a comparison of the ease of the methods, and the quality of graphics produced with different device drivers for each destination. Finally, we look at an additional method, the ODS Layout destination, which will be available for producing PDFs in Version 9 of SAS.

## **Graphic Device drivers**

Although this paper contains code for some neat looking graphics, the focus will be on the code required to place multiple graphics on one page. However, the choice of the proper graphics device driver is an important consideration when rendering graphics for output to either PDF or RTF documents. When choosing your device driver there are two elements that must be considered:

- 1) The complexity of the graphic
- 2) The destination for the graphic.

For this paper I have chosen particularly complex graphics, which will look best with a very high resolution graphics driver. To be honest, for most graphs (bar charts, line graphs, pie charts, etc.) that will be used in "run of the mill" reports, high resolution graphics are not a necessity. Most of the time a bar chart rendered as a GIF file will look just as good as one rendered as a PNG file, and both of those will be just as good as a CMG file. In this situation the destination for the graphic (PDF, RTF, HTML document, standalone graphic to be imported into PowerPoint later) is probably more important. Table 1 provides my recommendations for which device driver to use with each destination named above.

Table 1: List of suggested graphics device drivers for specific destinations

Destination	Device
HTML	JPEG, GIF, Java <sup>1</sup>
PDF	PDF, PDFC
RTF	PNG, CGMOF97L <sup>2</sup> ,
	CGMOF97P <sup>3</sup> ,
PowerPoint	CGMMPPA⁴

1 Produces Java enabled GIF file

2 CGM for Microsoft Office 97 (i.e. MS WORD) Landscape

3 CGM for MS Office 97 Portrait

4 CGM for MS PowerPoint

However, I would suggest that you try multiple drivers to see which produces the best graphic for your purpose. A list of all the Graphics drivers included as part of SAS/GRAPH software can be obtained by submitting:

proc gdevice catalog=sashelp.devices nofs; list \_all\_; run; quit;

You can then substitute a specific device name (e.g. PNG) for the keyword \_all\_ to get a detailed description of the particular device. You may also consider using the SAS GDEVICE procedure to try making your own device if necessary. In development of this paper, I used device=PDFC for the graphs to be rendered in PDFs and device=CGMOF97L for graphs going into RTFs. The latter was chosen over PNG in this particular case because I wanted the highest resolution possible for my graphs.

# Sending Graphics to PDFs

With the technical details of how to make graphs out of the way, lets get to the main point of the paper, how do we send graphs out to a document. First lets quickly review the syntax to export a graph to either the PDF or RTF destination. The basic syntax to export a graphic to either the PDF or RTF destination is:

ODS PDF (RTF) file="Path to your file.(correct extension for file created)"; SAS/GRAPH statements; ODF PDF (RTF) close;

For example:

data spiral; do theta=1 to 1080 by .1; r=.75 + sin((.25\*theta)); x=r\*cos(theta); y=r\*sin(theta); output; end; run;

options orientation=landscape; goptions reset=all device=pdfc /\*NOTICE my choice for the PDF destination\*/ ftext="Swissb";

ods pdf file="Your path.pdf"; proc gplot data=spiral; plot y\*x/ noaxes nolegend noframe; symbol1 interpol=join width=10 ci=cxEC8B16 cv=cx546EF0 font=swiss value=-; title h=5 "Graph Number 1 "; run;

# - 2 -

#### quit; ods pdf close;

will produce the graphic below (as a full page in a PDF document):



Figure 1: Spiral Graph rendered on a full page in a PDF document, using the PDFC graphics device

Notice above that I used the graphic device PDFC for color PDF graphics, I could use either the GIF or JPEG device with similar results, but, as discussed in the previous section, I have found that PDFC (perhaps understandably) works best for the PDF destination of the ODS.

To output to the RTF destination simply change the above code to:

- 1) change the graphics device (try the recommended device=CGMOF97L or PNG) and
- 2) Use ODS RTF (rather than PDF).

Now lets make three more graphs and add them to our PDF.

\*ods pdf close; \*Re-run the previous GPLOT and leave the PDF open to add more graphs;

#### data star;

do theta=1 to 1080 by .75; r=sin((5)\*theta); x=r\*cos(theta); y=r\*sin(theta); output; end; run;

proc gplot data=star;

plot y\*x/nolegend noframe noaxes; symbol1 interpol=join width=10 ci=cxEC8B16 cv=cx546EF0 font=swiss value=-; title h=5 "Star Power "; run; quit; Produces:



Figure 2: A "Star" graphic, rendered as Page 2 of a PDF document

data colors;

do theta=1 to 5760 by 1; r=sin((sqrt(3/2))\*theta); x=r\*cos(theta); y=r\*sin(theta); z=(round(rannor(0),.01)); if abs(z) gt 1 then z=1-abs(z); output; end; run;

#### %macro fixsymbols;

%\*Used to set symbol definitions for all 99 symbols, which will be used to make the multicolor graph below;

goptions reset=all ;\*used color and symbol definitions above, return to default before modifying here;

# %do i=1 %to 99;

symbol&i height=.5 v=dot i=join line=35; %end; %mend;

. . . ,

%**fixsymbols**;

goptions ftext="Swissb";

proc gplot data=colors; plot y\*x=z/nolegend noframe noaxes; title h=5 "Many colors "; run; quit;

Creates:



Figure 3: Multicolor sphere as Page 3 of our PDF

data rose; do theta=1 to 360 by .5; r=sin((9/2)\*theta); x=r\*cos(theta); y=r\*sin(theta); output; end; run;

proc gplot data=rose; plot y\*x/ nolegend noframe noaxes; symbol1 interpol=join width=10 ci=cxC01C6E cv=cxA7398A font=swiss value=-; title h=5 "Pretty flower "; run; quit;

ods pdf close;



Figure 4: A pretty flower on page 4 of our PDF

As you can see, this code produces a file that puts four graphs each on a separate page. It doesn't look bad, but there are many instances where we might want to compress this so that all four graphs are on a single page.

This is a relatively simple task in the PDF destination.

# Four graphs on one page, ODS PDF

To place multiple graphs on a single page in the PDF destination we can use two types of options:

- 1. the ODS options STARTPAGE=NEVER and,
- 2. SAS/GRAPH options that control the position of the graphs on the page.

These SAS/GRAPH options include HSIZE and VSIZE to control the size of the graph and HORIGIN and VORIGIN to control the position from which the graph starts on the page. STARPAGE=NEVER tells SAS not to go to a new page even at the beginning of a graphic.

Expanding on our example from above we would code:

goptions reset=all
device=pdfc
ftext="swissb";

ods pdf file="C:\Your Path\four in one.pdf" startpage=never; ods proclabel "Four graphs on one page"; goptions hsize=5.20 in vsize=3.75 in; title1;

# proc gplot data=spiral;

goptions horigin=0in vorigin=3.99 in; title2 'PLOT #1-A spiral'; plot y\*x /name="Spiral" noaxes nolegend noframe; symbol1 interpol=join width=10 ci=cxEC8B16 cv=cx546EF0 font=swiss value=-; run; quit;

## Proc gplot data=star;

goptions horigin=5.25; title2 'PLOT #2-A star'; plot y\*x /nolegend noaxes name="star" noframe; symbol1 interpol=join width=10 ci=cxEC8B16 cv=cx546EF0 font=swiss value=-; run; quit;

#### %fixsymbols;

proc gplot data=colors; goptions hsize= 5.20in vsize= 3.75in device=pdfc ftext="swissb" horigin=0 in vorigin=0 in ; "Need the above goptions because of the reset=all in the Macro fixsymbols; title2 'PLOT #3-Multicolor "thing"; plot y\*x=Z / name="Colors" nolegend noaxes noframe; run; quit; proc gplot data=rose; goptions horigin=5.25 in; title2 'PLOT #4-A rose'; plot y\*x / name="Rose" noframe nolegend noaxes; symbol1 interpol=join width=10 ci=cxC01C6E cv=cxA7398A font=swiss value=-;

**run**; quit;

ods pdf close;

will produce a PDF with all four graphics on one page.



Figure 5: a PDF document with four graphs on one page.

Note that you can change the number, size and position of the graphs by changing the graphics options (Hsize, Vsize, Horigin, and Vorigin) accordingly.

Unfortunately Startpage=Never does not work in the RTF destination, so that each graphic will end up on a different page, no matter how you try to resize and reposition them with Graphics options.

# An RTF solution

You can get four graphs onto one page in an RTF document; it just requires a little more ingenuity. The solution offered here uses PROC GREPLAY to combine the four separate graphs into one image within SAS and then uses DDE to import that image into MS WORD.

In order to produce the (high-resolution) graphs for this paper, it is best to create all the original graphs using the graphics option DEVICE=CGMOF97L, rather than PDFC as shown in the PDF example above.

The other thing we need to do is tell SAS to send the "Four in One" graph to a folder where we can find it later, we use the graphics option GSFNAME to do this. Adding the following statements in place of ODS PDF in the "Four in One" example above will take care of both of these problems.

- 4 -

%let path= C:\MY PATH; filename graphout "&path";

goptions reset=all device=cgmof97I gsfname=graphout

The only other modification to the PROC GPLOT statements we used in the PDF example would be to remove the GOPTIONS statements (e.g. HSIZE HORIGIN etc.) from the individual procedures, for the RTF destination, we will resize and reposition our graphs with PROC GREPLAY.

# PROC GREPLAY

The GREPLAY procedure is designed to accomplish a task similar to what we used the SAS/GRAPH options HSIZE VSIZE HORIGIN and VORIGIN for in our PDF example. It takes existing SAS/GRAPH images from a Graphics Catalog and places them in a template according to specifications you set. For example:

## proc greplay tc=work.tempcat nofs;

tdef newtemp des='Four panel template'

1/ulx=5 uly=95 urx=48 ury=95 llx=5 lly=52 lrx=48 lry=52 color=blue

2/ulx=52 uly=95 urx=95 ury=95 llx=52 lly=52 lrx=95 lry=52 color=red

3/ulx=5 uly=48 urx=48 ury=48 llx=5 lly=5 lrx=48 lry=5 color=orange

4/ulx=52 uly=48 urx=95 ury=48 llx=52 lly=5 lrx=95 lry=5 color=green

Above makes the template, below uses it;

template newtemp;

igout work.gseg; gout work.gseg;

treplay

1:Spiral 2:Star 3:colors 4:rose name="Fourin\_1";

quit;

The above example is divided into two sections, separated by the comment (\*\*\*\*\*). Above the line I create a template that has four equal spaces for the graphs on a page and distinguish each graph by creating a colored border around them. You identify

sections of the template with numbers, in this case 1-4. Then you create sections of the template by specifying the four corners of each area of the new graphic that will accommodate a previously created graphic. For instance:

1/ulx=5 uly=95 urx=48 ury=95 llx=5 lly=52 lrx=48 lry=52 color=blue

describes section number 1 of the graph. ULX describes the Upper left limit for the X (horizontal) corner of the section in the template. In this case I start this section at horizontal position 5 (ranging 0-100 from left to right on a page), and vertical position 95 (ranging from 0-100 from bottom to top on a page), or the top left corner of the document.

This section extends from the top left corner to 48% of the way across (X=48) and 48% of the way down (Y=52\%).

ULX, ULY	URX,URY
LLX,LLY	LRX,LRY

Figure 6: Position of the X and Y coordinates that make up the four corners of a section in a Graphics Template.

Figure 6 shows the position of each X and Y component used to describe the sections in the template. Each of the other sections in this template corresponds to a box in one of the other three corners of the page.

The second part of the code above (below the comment (\*\*\*\*\*) line), uses the template that was created. The TREPLAY statement specifies which existing graphs go in which section and optionally allows you to give a name to your graph.

In this example the graph will be stored in our GSFNAME folder as fourin\_1.cgm.

# Inserting the graphic into Word

You can move the graphic created via PROC GREPLAY into WORD by surrounding the procedure with ODS RTF statements. However, the default size of a graphic created with the CGMOF97L graphics device appears to be about 3 inches X 4 inches. Assuming we want our Four graph template to take up the entire page in WORD, we have to resize the graphic. This means we will most likely want to resize our graphic in WORD, and to do this we need to use DDE. Since we will have to use DDE anyway, it seems to make sense to use it to insert the graph into WORD, so we are in complete control of the process.

There are a couple pieces of housekeeping that need to be done in SAS before we can start sending DDE commands to WORD.

options noxsync noxwait xmin; \*Important options to set when using DDE;

filename wordsys dde "winword|system"; \*Set up the conversation between Word and SAS;

\*The code below (borrowed from one of the many excellent DDE papers by Koen Vyverman) Checks to see if Word is open already, and opens it if it is not.;

data \_null\_; length fid rc start stop time 8;

- 5 -

fid=fopen('wordsys','s'); if (fid le 0) then do; rc=system('start winword'); start=datetime(); stop=start+10; do while (fid le 0); fid=fopen('wordsys','s'); time=datetime(); if (time ge stop) then fid=1; end; end: rc=fclose(fid); run;

Ok, now we are ready to use SAS to tell Word that it wants our Graph, and then tell WORD how we want the graph to look. How do we do this??

- Open a new (blank) Word document 1)
- Set the Margins of the document to be .25inches on all 2) sides and
- 3) Turn the document to Landscape orientation
- Insert our Picture (the Four\_in\_One graph) 4)
- 5) Center the Picture and
- Resize it to take up the whole page 6)

\*The code here represents the DDE commands that actually insert our graphic and manipulate it in Word;

#### data \_null\_; file wordsvs:

put '[FileNewDefault]'; \*1) Open a blank Word document. WORD must be open already!!;

put '[Filepagesetup .TopMargin=".25 in" .bottommargin=".25 in" .leftmargin=".25 in" .rightmargin=".25in" ]'; \* 2) Fix the Margins; put '[Toggleportrait]'; \*3) Turn page orientation to landscape;

- put '[InsertPicture ""&path.\fourin\_1.gif"""]'; \*4) Inserts our graph; put '[Editgoto "g"]';
- put '[CharRight 1,1]'; \*5a) Selects the Graph;
- put '[Centerpara]'; \*5b) Centers it;
- put '[FormatPicture .scaleX="250%" .scaleY="250%"]';

6) based on default CGM size (as imported into MS WORD) of approximately 3 inches by 4.00 inches this fits perfectly!!; put '[FileSaveAs .Name = "" "&path.\graphout.doc" "" .format=0]';

\*Save our new file; put '[Fileclose]';\*if done;

run;

After Running this code we get a file named Graphout.Doc in our "PATH" folder, with a graphic that looks something like this:



Figure 7: Print Preview of Graphout.doc, our MS WORD file with four graphs on a page

Although you can't really tell from this view, the CGMOF97L graphics device actually produces a much higher-resolution graphic than the PDFC graphic device. If you try the code from this paper at home you will find the CGM file in WORD should have much prettier "petals" on the rose graph, and a much more

detailed structure to the multi-color oval. Again, most of the time you won't be producing pictures like this from SAS/GRAPH, but I want to emphasize one more time that you should try a couple different graphics devices to see which one produces the best output for your given task.

## Coming to a Version 9 SAS session near you!

I would be remiss if I didn't mention that Version 9 of SAS will offer one more way to send Multiple graphics to a single page in the PDF (or other PRINTER) destination, ODS LAYOUT. There are two ways to use ODS LAYOUT, with Absolute Layout or Relative layout. Both types of LAYOUT work by dividing a page into regions into which SAS places output.

Absolute layout functions very similarly to PROC GREPLAY, in that you identify regions by telling SAS where to start a region using X and Y coordinates, and then specify the size of the region as percentages of the total horizontal and vertical space on a page.

For Example,

data test; input x y @@; datalines: 1 2 2 4 3 8 4 10 5 15 run;

title2"NEED TITLE BEFORE ODS PDF STATEMENT";

ods pdf file="&path.\test ODS Layout - Absolute.pdf";

ods layout Start width=10in height=8in ;

ods region x=0 y=5% width=45% height=45%; proc gplot data=test; title2 'PLOT #1': plot y\*x /name="mygraph1" noframe; run;

ods region x=55% y=5% width=45% height=45%; title2 'PLOT #2'; plot y\*x /name="mygraph2" noframe; run:

ods region x=0 y=51% width=45% height=45%; title2 'PLOT #3': plot y\*x / name="Mygraph3" noframe; run<sup>.</sup>

ods region x=55% Y=51% width=45% height=45%; title2 'PLOT #4'; plot y\*x / name="Mygraph4" noframe; run:

#### quit;

ods layout end; ods pdf close;

creates a PDF with four graphs on a page like the one in Figure 8. Notice that ODS LAYOUT starts (quite logically) with an ODS LAYOUT START statement, on which you define the size of the page you are working with using the WIDTH and HEIGHT options. You then use ODS REGION statements before each piece of output to place the output in a section of the page. The upper left corner of the REGION is defined by X=Percentage and Y=Percentage, which would correspond to ULX and ULY in PROC GREPLAY as described above. You also specify a percentage of the total width and total height to define the size of the region. Finally, you end your LAYOUT definition with an ODS

- 6 -

LAYOUT END statement, prior to closing your ODS destination (ODS PDFCLOSE;).



# Figure 8: A PDF with four graphs on a page created using ODS Layout

You can also use relative layout in a similar fashion. For relative layout, you divide the output into columns and rows, and then tell SAS which output to place in which Column/Row location.

For Example:

ods PDF file="&path.\test ODS Layout - Relative.PDF";

ods layout Start width=10in height=8in columns=2 rows=2 column\_gutter=.5in row\_gutter=.8in row\_heights=(3.5in 3.5in);

ods region row=1 column=1; proc gplot data=test; title2 'PLOT #1'; plot y\*x /name="mygraph1" noframe; run;

ods region row=1 column=2; title2 'PLOT #2'; plot y\*x /name="mygraph2" noframe; run;

ods region row=2 column=1; title2 'PLOT #3'; plot y\*x / name="Mygraph3" noframe; run;

ods region row=2 column=2; title2 'PLOT #4'; plot y\*x / name="Mygraph4" noframe; run;

## quit;

ods layout end; ods PDF close;

For this version of ODS LAYOUT, there are many more options that need to be specified on the ODS LAYOUT START statement. In addition to Width and HEIGHT which we saw in the Absolute layout example, we specify:

- 1. The number of columns (columns=N) and
- 2. The number of rows (rows=N)
- 3. The space between columns (column gutter=)
- 4. The space between rows (row\_gutter=)
- The Height of the rows (row\_heights=(row1height, row2height, etc)

After setting up our matrix of columns and rows, the ODS REGION statements simply identify the Column/Row destination for each piece of output. And, just like Absolute LAYOUT, Relative LAYOUT statements end with an ODS LAYOUT END; before closing the ODS destination.

I should point out that ODS LAYOUT is still experimental in Version 9, and refer you to Brian Schellenberger's paper on the SAS website for more details on current ODS LAYOUT syntax, and future directions for this resource.

Unfortunately, ODS LAYOUT is only supported for the PRINTER family of ODS destinations, so it will work when creating a PDF document, but won't work for an RTF.

## Conclusion

This paper provides yet another example of how many different ways there are to accomplish the same task in SAS. I hope I have shown you enough different ways of creating documents with multiple graphs on a page to peak your interest, and point you in the right direction. I think the suggestions for how best to create these multi-graph documents are a good foundation, and they are presented to give guidance as to which procedure will produce the best High-resolution output for your destination of choice. However, they are just suggestions, and I hope you see them as building blocks that you can use to develop your own solutions for creating reports with many good-looking graphs.

## References

#### SAS/GRAPH:

SAS Institute Inc. (1999) SAS/GRAPH Software: Reference, Version 8. Cary, NC SAS Institute, Inc.

#### ODS:

SAS Institute, Inc. (1999) The complete guide to the SAS Output Delivery System, Version 8. Cary, NC SAS Institute, Inc.

#### SAS Version 9 ODS LAYOUT:

Schellenberger, Brian (2002): New ODS Printer features for V9 <<u>http://www.sas.com/rnd/base/topics/odsprinter/new9.html</u>> (December 13, 2002)

#### DDE:

SAS Institute, Inc. (1999) SAS Companion for the Microsoft Windows Environment, Version 8. Cary, NC SAS Institute Inc.

Vyverman K, Viergever W. (2003) Fancy MS Word Reports Made Easy: Harnessing the Power of Dynamic Data Exchange. Proceedings of the Twenty-Eighth Annual SAS® Users Group International Conference, March 2003.

Any other SUGI papers by Master Vyverman. <<u>http://www.vyverman.com/professional/papers.html</u>> (December 2002)

## **Contact Information**

Kevin P. Delaney MPH Northrop-Grumman Mission Systems Atlanta, GA KDelaney@cdc.gov

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 registered trademarks or trademarks of their respective companies.