

Paper 222-2008
Multiple Graphs on One Page Using SAS/GRAPH® V9
Hany Aboutaleb, Biogen Idec, Cambridge, MA

Abstract:

This paper will reveal that like good writing, producing an effective graphical display requires an understanding of the data you want emphasize. SAS/GRAPH® V9 can be used in combination with the Output Delivery System (ODS) to produce high-resolution multiple graphs on one page. Achieving multiple graphs on one page effect can be challenging. However, like almost every SAS® solution there are multiple ways you can achieve this goal with SAS/GRAPH® V9. This paper will compare and contrast three different methods for achieving this result.

Introduction:

In addition to traditional SAS/GRAPH® PROC GREPLAY, and Data Step Graphics Interface (DSGI), SAS has added ODS LAYOUT capability to SAS/GRAPH® V9 to programmer's tool set. All of these methods can deliver big payoffs when generating multiple graphs on one page. SAS/GRAPH® is designed to produce high quality graphical representations of your data and can run on all platforms and can display on many devices and many formats. The choice of a proper graphics device driver is an important consideration when producing graphs.

Multiple graphs on one page:

It is useful to display multiple graphs on one page to see patterns and relationships. To place multiple graphs on a single page you can:

1. Create the graph and store it in SAS catalog and use PROC GREPLAY.
2. Use the Data Step Graphics Interface (DSGI).
3. Use the new ODS LAYOUT.

In my examples, I output the graphs to the default graphics catalog of GSEG. You can also specify a different name by using the GOUT option on the PROC statement. If you are running your code multiple times in the same interactive session, it is often useful to delete the earlier attempts to have more control over the graph names. By using PROC GREPLAY you can delete the existing graphs programmatically. You can easily put multiple graphs on one page; however, you need to use additional options like the NODISPLAY option which tells the graph to write the graph to the catalog.

To delete earlier graphs from your catalog (GSEG) use:

```
proc greplay igout= GSEG nofs;  
delete _all_;  
run;  
Quit;
```

Option to control the graph:

```
goptions reset=all nodisplay gsfmode=replace;
```

Once you have created all graphs, you will need to reset the options to display the graph and reset the size to full size in order to insert the desired graphs on one page.

PROC GREPLAY:

This PROC does what it says - it replays your existing graphs. It takes existing SAS/GRAPH® images from a graphics catalog and places them in a template according to the specifications you set.

Example 1: Place the 4 graphs in a template:

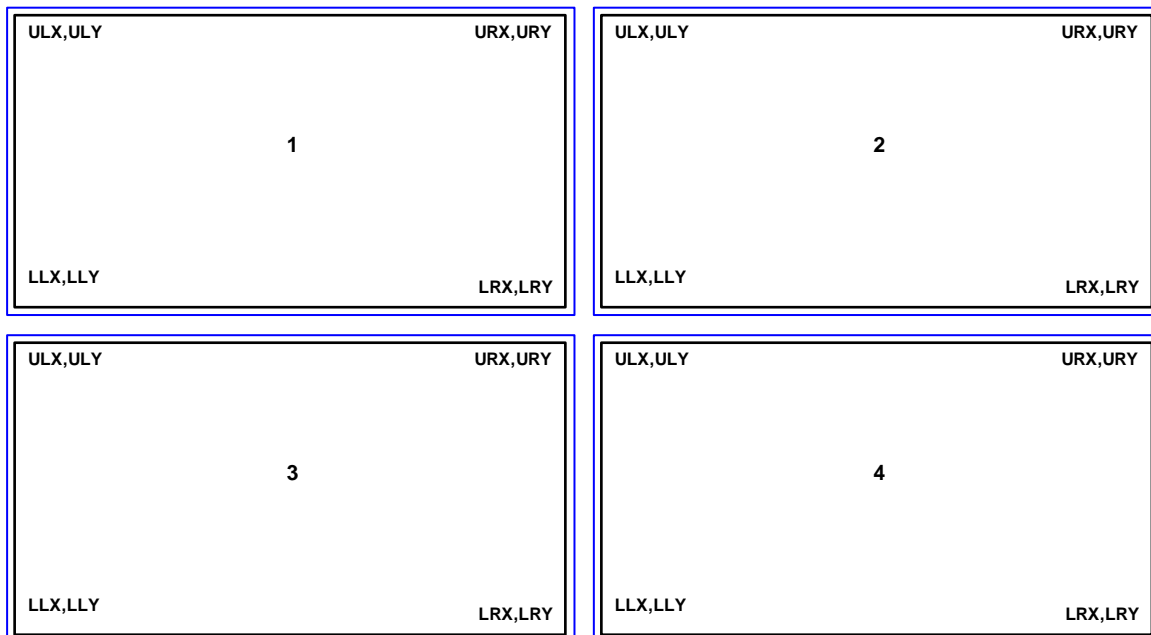
```
proc greplay tc=work.tempcat nofs;
```

```
tdef newtemp des='Four panel template'
```

<pre>1/ulx=5 uly=95 urx=48 ury=95 llx=5 lly=52 lrx=48 lry=52 color=blue</pre>	<pre>2/ulx=52 uly=95 urx=95 ury=95 llx=52 lly=52 lrx=95 lry=52 color=red</pre>	<pre>3/ulx=5 uly=48 urx=48 ury=48 llx=5 lly=5 lrx=48 lry=5 color=orange</pre>	<pre>4/ulx=52 uly=48 urx=95 ury=48 llx=52 lly=5 lrx=95 lry=5 color=green;</pre>
---	--	---	---

```
run; quit;
```

First 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. For instance, in 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 start this section at horizontal position 5 (range 0-100 from left to right on a page), and vertical position 95 (range 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%).



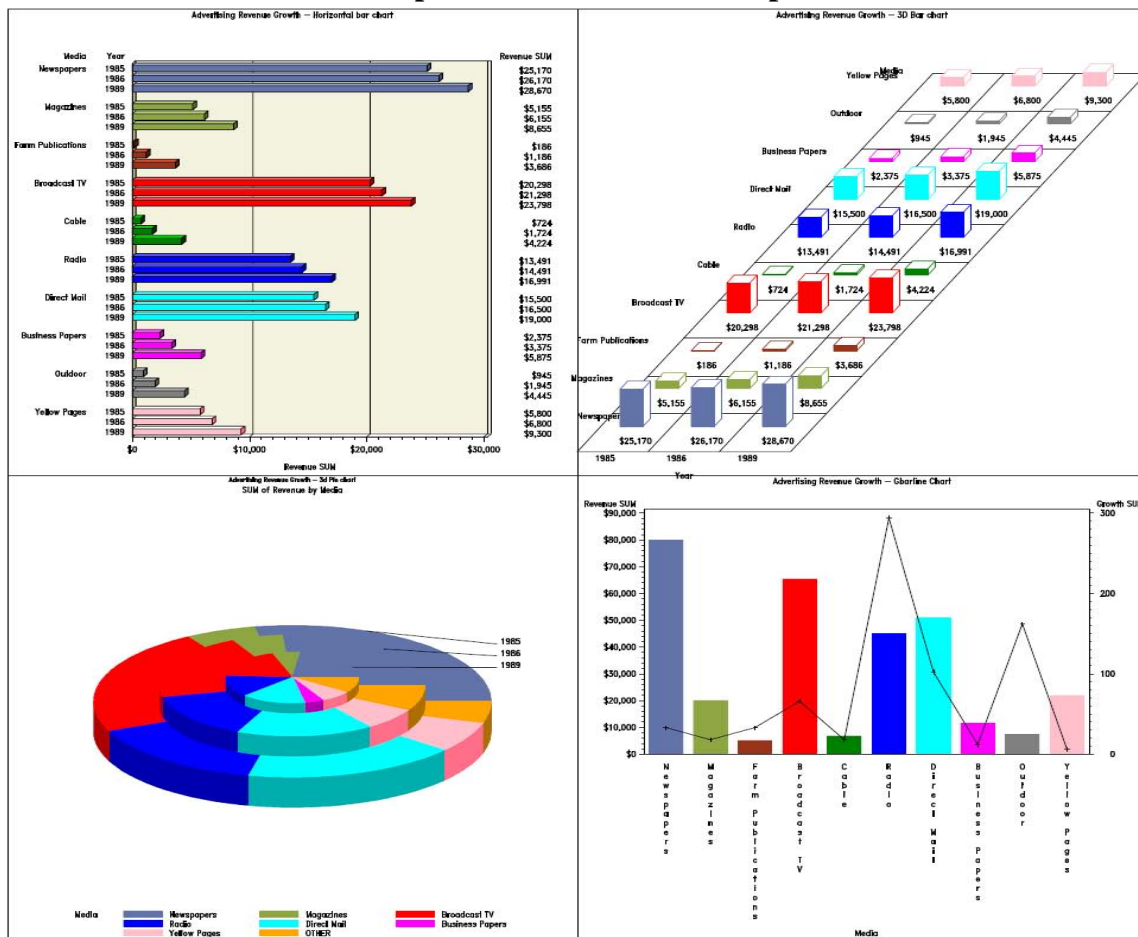
Example 2: Using the built-in template to insert 4 graphs.

In addition to building your own template, SAS provides several built-in templates. The TC or template catalog option points to the built-in templates that are found in SASHELP.TEMPLT. The IGOUT option specifies the graphics catalog to use for the input graphs. The TEMPLATE statement shows the name of the template. L2R2 is 2 boxes left and 2 boxes right. You can click on VIEW in the catalog to find the description and coordinates of each template. The TREPLAY statement shows which graph to put in which box based on the names of the graphs. This is why specifying the name of the graphs is important.

```

goptions display hsize=8 in vsize=10 in;
proc greplay igout= GSEG nofs tc=sashelp.templt;
    template l2r2;
    treplay 1:hany1 2:hany2 3:hany3 4:hany4;
run;
quit;
    
```

Example 2 Proc GREPLAY output

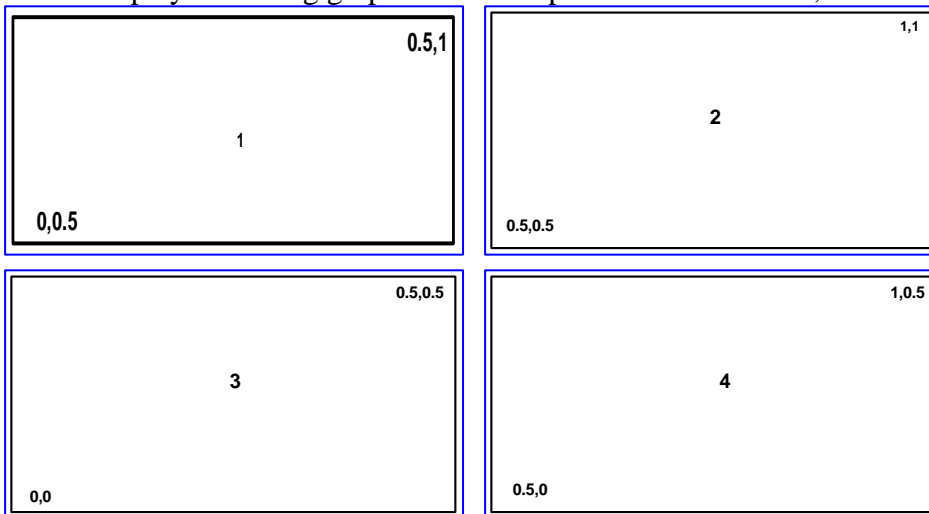


DATA Step Graphics Interface (DSGI):

The DATA Step Graphics Interface (DSGI) is a component of SAS/GRAPH® software that enables you to generate graphics output from within a DATA step, macro or SCL application. Through DSGI, you can call or add features to an existing graph generated by SAS procedure PROC and create a customized or a drill-down graph. It is easier to add common titles or footnotes within DSGI than within GREPLAY, since title and footnote do not work in the GREPLAY procedure without GSLIDE. With Viewport and Window functions, DSGI enables you to display an existing graphic in a specified area. The default Viewport is defined as (0, 0) to (1, 1) while 1 represents 100 percent of the graph output area. Once the Viewport is defined, you can decide which part of an existing graph is displayed by defining Window from coordinates (0, 0) to (100, 100). For example, if you do not want to display the titles generated by the previous procedure PROC, you can define smaller Window coordinates to cut off the titles. Before inserting an existing graph with the Gset function, which defines Viewport and Window, you have to make a transformation to activate both Viewport and associated Window. DSGI has 21 transformations from 0 to 20. Transformation 0, by default, always uses the entire output area for Viewport and fills in the Viewport with the graph, which is further defined by the Window Coordinates. For example, if you like to separate the entire output area in 4 specified areas, you need to make transformations 4 times to activate each Viewport and associated Window, respectively.

Example 3: Display 4 existing graphs in one output area with DSGI .

Title1 "Display 4 existing graphs in one output area with DSGI";

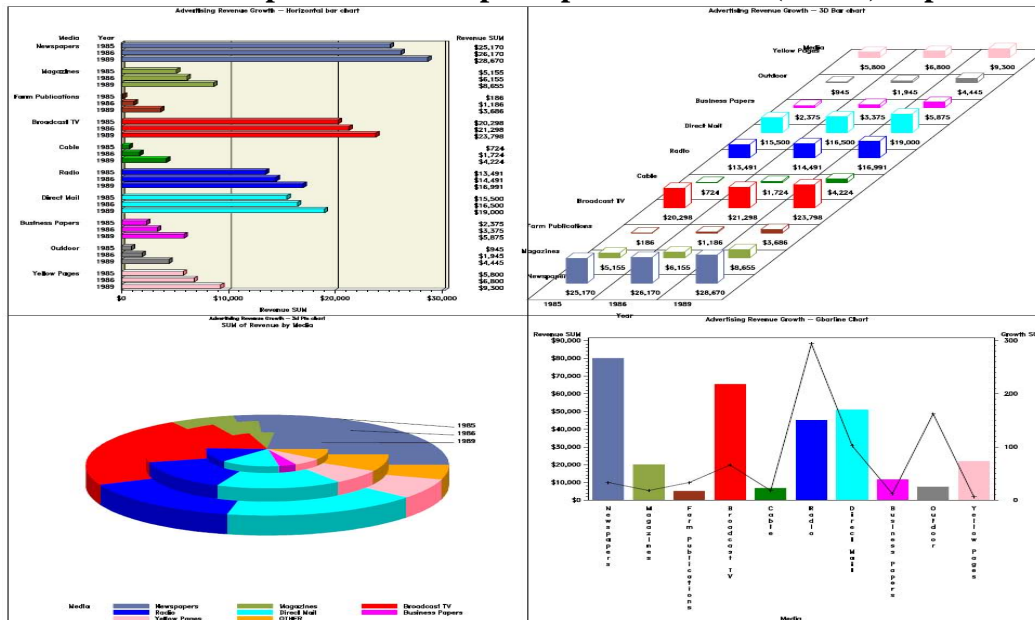


```
Data _null_;
/*Initialize DSGI*/
DSGI = Ginit ();
/*Open a graphic segment*/
DSGI = Graph (.Clear');
*** Define first Viewport as (0, 0.5) to (0.5,1) ***;
DSGI = Gset ('Viewport', 1, 0, 0.5, 0.5, 1);
*** Define associated Window coordinates as (0, 0) to (100, 100) ***;
```

```

DSGI = Gset ('Window', 1, 0, 0, 100, 100);
*** Transformation 1 activates Viewport and associated Window ***;
DSGI = Gset ('Transno', 1);
*** Insert an existing graph generated by previous PROC procedure with Name=Han1 *;
*** Define 2nd Viewport as (0.5, 0.5) to (1,1) ***;
DSGI = Graph ('Insert', 'Han1');
DSGI = Gset ('Viewport', 2, 0.5, 0.5, 1, 1);
DSGI = Gset ('Window', 2, 0, 0, 100, 100);
DSGI = Gset ('transno', 2);
DSGI = Graph ('Insert', 'Han2');
*** Define 3rd Viewport as (0, 0) to (0.5,0.5) ***;
DSGI = Gset ('Viewport', 3, 0,0, 0.5, 0.5);
DSGI = Gset ('Window', 3, 0, 0, 100, 100);
DSGI = Gset ('transno', 3);
DSGI = Graph ('Insert', 'Han3');
*** Define 4th Viewport as (0.5, 0) to (1,0.5) ***;
DSGI = Gset ('Viewport', 4, 0.5, 0, 1, 0.5);
DSGI = Gset ('Window', 4, 0, 0, 100, 100);
DSGI = Gset ('transno', 4);
DSGI = Graph ('Insert', 'Han4');
/*Close the graphics segment*/
DSGI = Graph ('Update');
/*Terminate DSGI*/
DSGI = Gterm ();
/*Display output*/
Run;
    
```

Example 3 DATA Step Graphics Interface (DSGI) output



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.

Example 4: Absolute layout

```

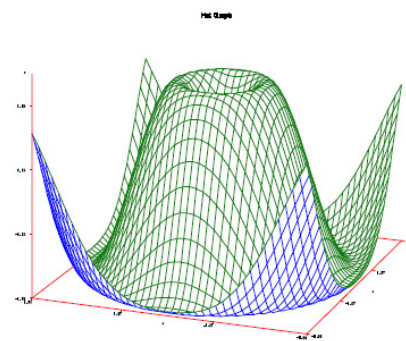
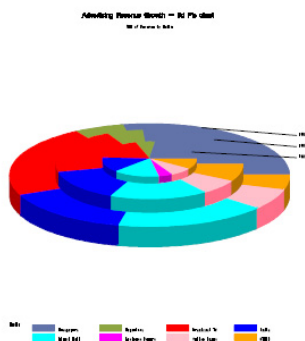
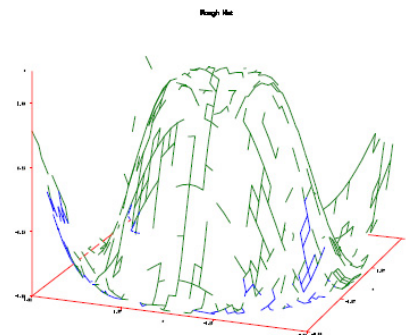
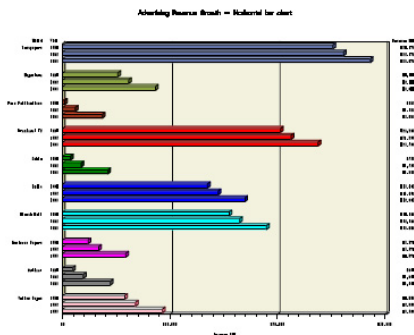
Filename odsout 'c:\';
title2 "Absolute layout";
ods pdf body="Absolute.pdf" ;
ods layout Start width=10in height=8in ;
ods region x=0 y=0 width=40% height=40%;
proc gplot data=test;
title2 'PLOT #1';
plot y*x /name="hany1" noframe;
run;
ods region x=50% y=0 width=40% height=40%;
title2 'PLOT #2';
plot y*x /name="hany2" noframe;
run;
ods region x=0 y=50% width=40% height=40%;
title2 'PLOT #3';
plot y*x / name="hany3" noframe;
run;
ods region x=50% Y=50% width=40% height=40%;
title2 'PLOT #4';
plot y*x / name="hany4" noframe;
run;
quit;
ods layout end;
ods pdf close;

```

This example will create a PDF with four graphs on a page like the one in ODS Layout-Absolute figure. 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 (ODS PDFCLOSE ;).

ODS Layout - Absolute layout

16:28 Monday, September 17, 2007 1



Example 5: Relative 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.

```
Filename odsout 'c:\';
ods PDF body="Relative.PDF" ;
```

```
ods layout Start width=10in height=8in columns=2 rows=2 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';
```

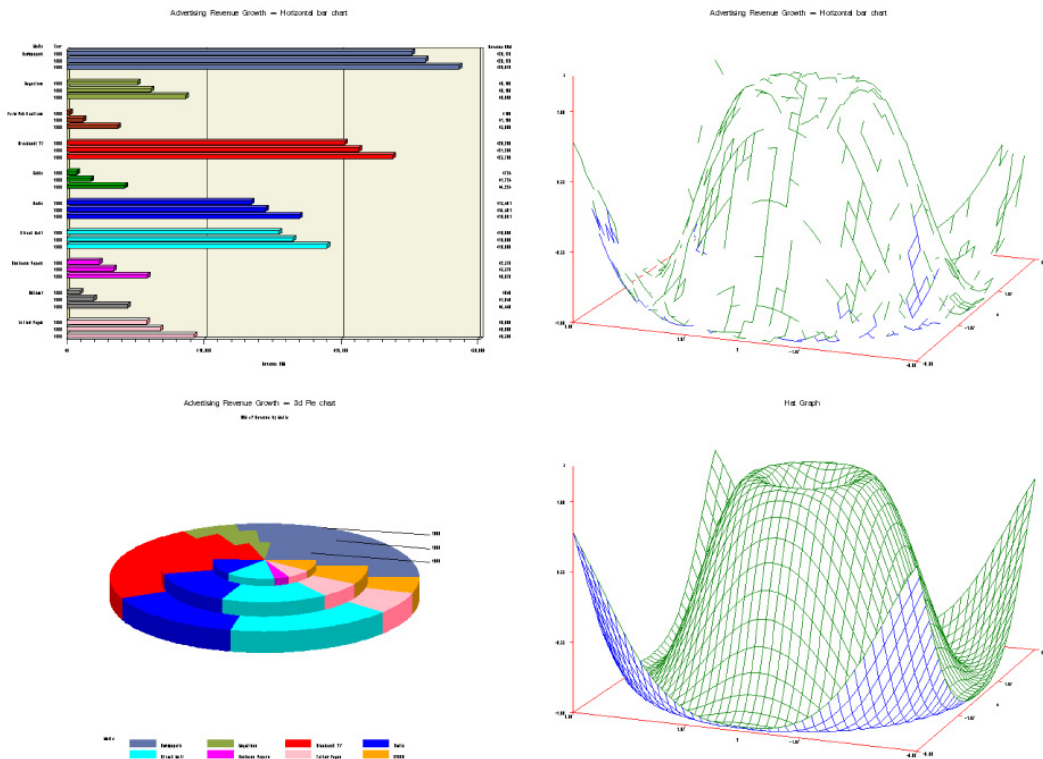
```

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;

```

ODS Layout - Relative layout

16:28 Monday, September 17, 2007 1



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)

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 V9.1.3.

Conclusion

This paper provides the readers with many different ways to accomplish the same task in SAS/GRAPH® V9 like almost every SAS® solution. I hope you can use those examples to generate multiple graphs on one page and develop your own solutions for creating page with many good-looking graphs.

Acknowledgment:

The author would like to thank Ms. Tina Casteris for her review and valuable comments.

References

- [1] SAS/GRAPH *Software: Introduction, Version 6, First Edition*, SAS Institute Inc. , Cary, NC
- [2] SAS/GRAPH *Software: Reference, Version 6, First Edition, Volume 1 and volume 2*, SAS Institute Inc. , Cary, NC
- [3] SAS 9.1.3 Output Delivery System: User's Guide; *SAS® 9.1.3 Output Delivery System: User's Guide, Volumes 1 and 2*

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.

Contact Information

Your comments and questions are valued and encouraged.

Contact the author at:

Hany Aboutaleb

Biogen Idec.

14 Cambridge Center

Cambridge MA 02142

Tele. #: (617) 914-7125

Fax: (617) 679-3280 Internet: hany.aboutaleb@biogenidec.com