

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

Dirk Spruck, Accovion, Marburg, Germany

ABSTRACT

Creating multiple graphs on one page has been a challenge in SAS before version 9.2. However, even in earlier SAS versions there is an easy and elegant way to create multiple graphs on one page without facing the problems of distorted fonts and axes.

The method presented in this paper uses PROC GREPLAY to overlay graphs. Individual graphs are created using SAS graphics procedures and stored in a SAS graphics catalog. When the individual graph is produced the axes are scaled using the LENGTH parameter of the AXIS statement. The graphs are also placed appropriately on the page using the ORIGIN parameter of the AXIS statement. In the next step templates are defined with PROC GREPLAY for display of the individual graphs. But instead of defining a grid with scaled templates all templates use 100% of the output area. The individually stored graphs are then simply overlaid to create one output with multiple graphs on one page.

The beauty of this approach is that the graphics procedure takes care of the scaling. This prevents the graphs from being distorted when PROC GREPLAY is used to place them in a scaled template. Another advantage is that we are not limited to the output of a single graphics procedure but can combine output from different procedures like GPLOT and GCHART. The AXIS statement gives full and transparent control over size of the individual graphs as well as the placement of the graphs in the output area.

INTRODUCTION

The template functionality in PROC GREPLAY was designed to enable the user to combine multiple graphs and place them in a certain location of the output area of a page. However, creating one page of output with multiple graphs (**Figure 1**) can be very tedious. The process described in the SAS/Graph manual involves defining an output template using the GREPLAY procedure. This output template is then “filled” with graphs stored in a SAS catalog. Usually this involves that the graphs are scaled down to fit in the cookie cutter sections of the template. The scaling of the graphs can result in illegible fonts and problems with the lengths of the axis.

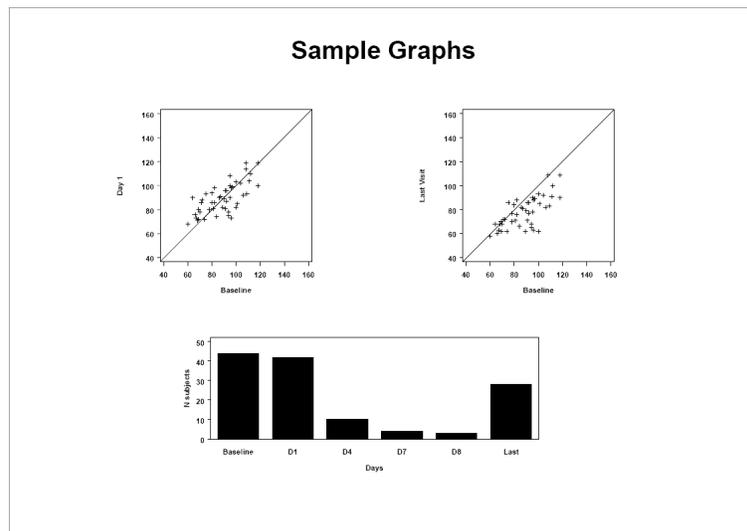


Figure 1: Example for multiple graphs combined on one page

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

The method presented in this paper was developed after trying to create twelve shift graphs on one page. Shift graphs are special because x and y axis should have the same length as they present the same range of data (**Figure 2**). However following the SAS manual it seemed nearly impossible to design GREPLAY templates in a way that the shift graphs still had equal lengths after scaling. In addition to the axis problem the proportion and size of fonts became distorted.

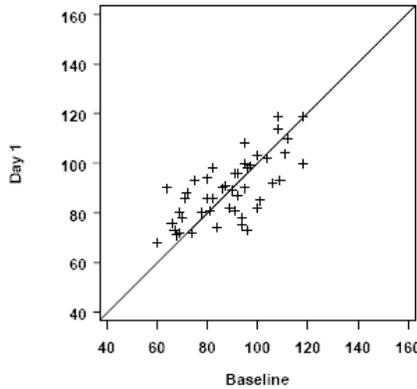


Figure 2: Example of shift graph with same length axes

The idea of this paper is to avoid the scaling of graphs as this is the main culprit for all sorts of problems. This can be done by creating “100%-templates”. Don’t worry if you never heard about this kind of templates because this term was made up in lack of a better name. 100%-Templates are GREPLAY templates that cover the full 100% of the output area. In other words, instead of having the output area divided into e.g. four equally spaced sections with 25% of the output area there will be four 100%-templates on top of each other (**Figure 3**).

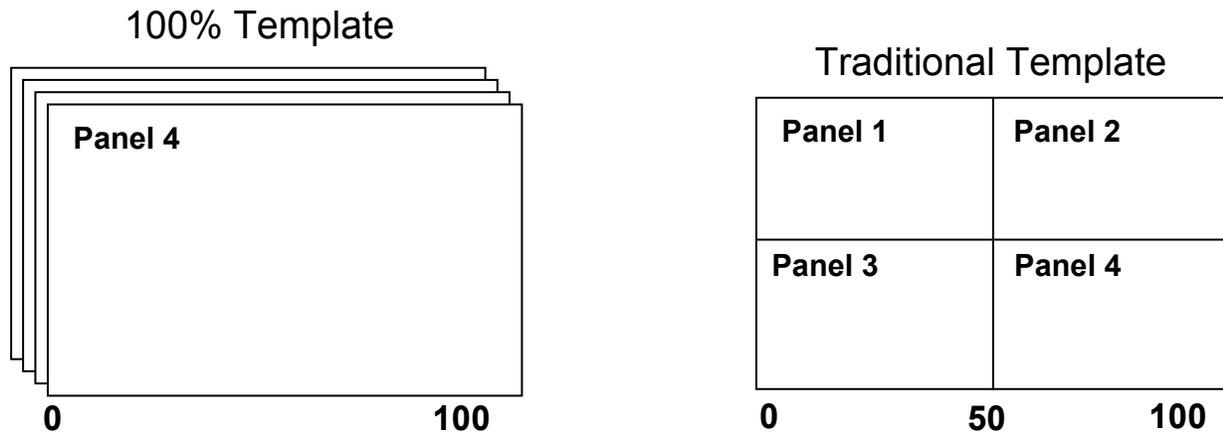


Figure 3: Example for a 100% template and the traditional GREPLAY template

Placing the graph in a certain location on the page can easily be achieved using the AXIS statements when the graph is created with one of the graphics procedures.

TRADITIONAL METHODS

In a SAS Global Forum 2008 paper Hany Aboutaleb provides a good introduction to different methods used to place multiple graphs on one page [1].

The first method introduced in Hany’s paper is the traditional method of using GREPLAY templates which we will discuss further below as it is the basis for understanding 100%-templates.

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

The second method discussed by Hany is the SAS DATA Step Graphics Interface (DSGI), which is a whole set of language elements for creating graphs. Although or better because DSGI is a very powerful tool it is very different from the “normal” SAS language most of us are used to.

The third method discussed in Hany’s paper uses ODS LAYOUT, which is available starting in SAS version 9. ODS LAYOUT lacks the backwards compatibility and is also restricted to ODS output, e.g. PDF or RTF. ODS LAYOUT works very similar to PROC GREPLAY and is certainly worth investigating.

With version 9 SAS has started developing ODS Graphics Template Language, which ultimately make the graphics interface flexible and powerful enough to deal with multiple graphs on one page. Introductions to ODS GRAPHICS and the Graphics Template Language can be found in presentations by Phil Holland [2] and SAS [3].

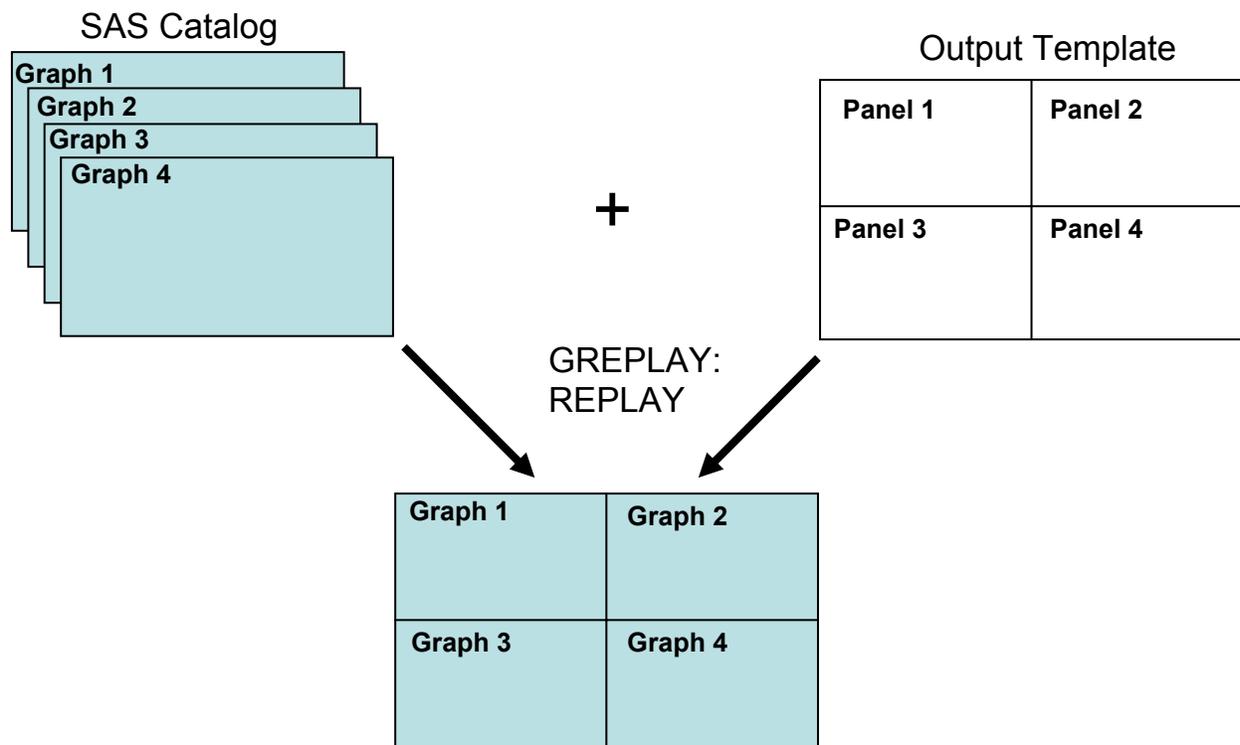
In their presentation at the Philadelphia SAS User Group Meeting in November 2007 Mary Buckwalter and Terek Peterson shared their frustration of finding a good solution for creating multiple graphs on a single page [4]. They propose using the overlay functionality of PROC GPLOT in combination with PROC GSLIDE and PROC GREPLAY to combine multiple graphs on one page. This is limited to the output of a single procedure. However the idea of overlaying graphs is key to the 100% template approach.

PROC GREPLAY

According to the SAS online documentation “the GREPLAY procedure displays and manages graphics output that is stored in SAS catalogs. The GREPLAY procedure also creates templates [...] that you can use when you replay your graphics output.” Traditionally by defining a template the output area is partitioned into different output sections. When the graphs stored in SAS catalogs are replayed using GREPLAY they are assigned to one of the sections.

The following illustration provides a quick introduction to the three steps necessary to combine four graphs on one page using the GREPLAY procedure.

- Step 1: Storing individual graphs in a graphics catalog
- Step 2: Creating a template with four sections
- Step 3: Replay stored graphs and assign them to the different sections



Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

Figure 4: Assigning graphs to output template

The GREPAY procedure is complex and only part of the functionality is covered by this paper.

USING PROC GREPLAY WITH “100%-TEMPLATES”

The process of combining multiple graphs on one page using the “100%-Templates” involves 3 steps, too:

Step 1: Create and store graphs and titles in a catalog using graphics and GSLIDE procedures.

Step 2: Define templates using the GREPLAY procedure

Step 3: Overlay graphs using the GREPLAY procedure

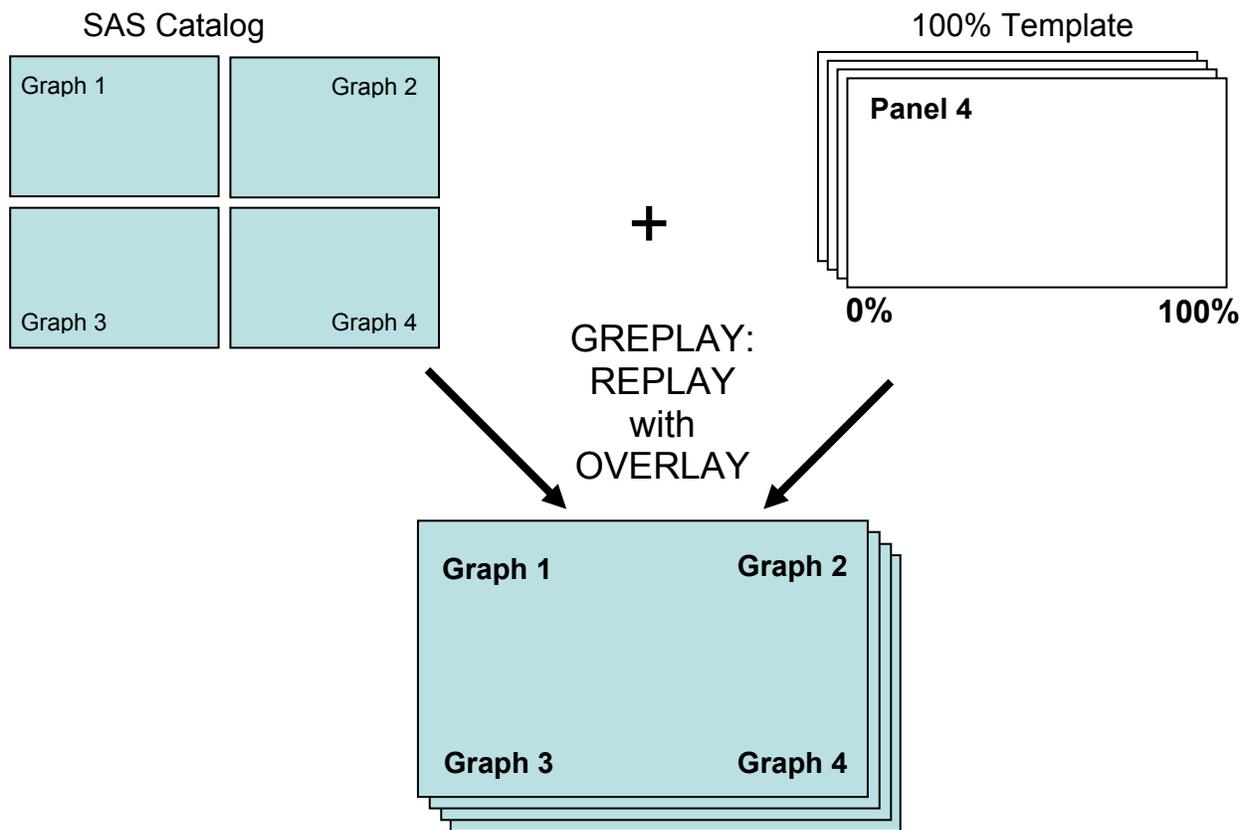


Figure 5: Creation of graphs (Step 1) and templates (Step 2) and the overlaid graphs (Step 3)

This process is in fact a modified combination of the three traditional methods steps 1 to 3 described above. The three steps of the new process are described in more detail below.

STEP1: CREATING AND STORING GRAPHS

The SAS/GRAPH procedures, e.g. GPLOT and GCHART can be used to create individual graphs. The graphs need to be stored in a SAS catalog in order to use them with GREPLAY later in the process. Deleting old graphs in the catalog and explicit naming of individual graphs helps to avoid using the wrong version of a graph.

When the traditional GREPLAY method is used the graphs need to be designed to fill as much of the page as possible (Figure 6) because they will shrink considerably when they are placed in the panels.

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

When graphs are created with Gplot or Gchart for use with the “100%-template” the graphs need to be already positioned and scaled. This can be achieved by defining and assigning AXIS statements.

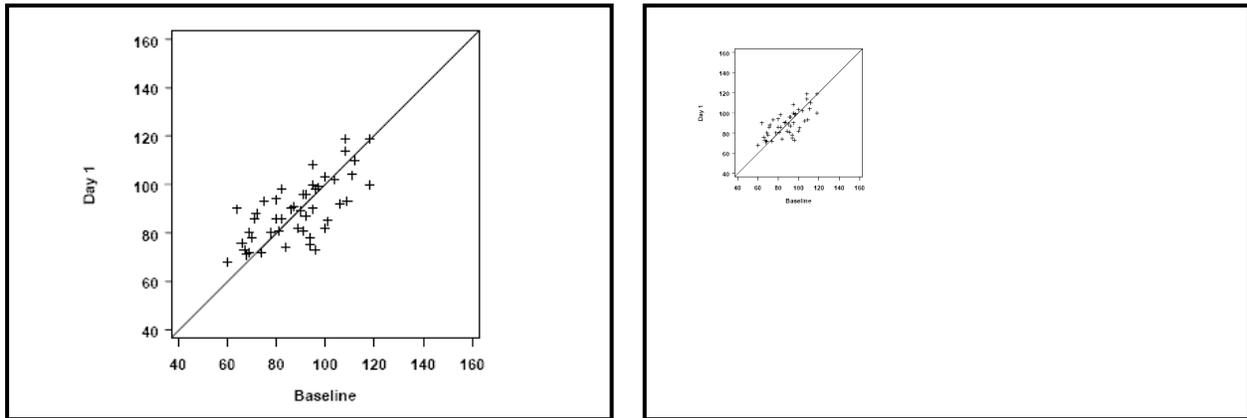


Figure 6: Graph filling most of the page (left) compared to scaled graph placed in upper left corner of output area.

The following sample code can be used to create a scaled and positioned graph, i.e. in this case a plot, and store it in a graphics catalog.

```
axis1 origin= (&x., &y.)cm length= 4cm ...;❶
axis2 origin= (&x., &y.)cm length= 4cm ...;

proc gplot data= plotdata gout= work.example;❷

    plot y* x/ haxis= axis1 vaxis= axis2❸ name= "plot1"❹;

run;
quit;
```

❶ **AXIS:** The ORIGIN statement within the axis definition is used to position the graph in the output area. The LENGTH statement is used to control the length of the axes. Usually the origin values for the both axes are the same. The length of the axis in this example is the same for both axes because a shift graph is created. The axis statement gives the programmer full control over the final appearance of the graph. Even after the following process the graph will exactly look as specified in the axis statement, i.e. the length of the axis will be 4 cm.

❷ **GOUT:** The GOUT statement will store the graph in a graphics catalog. Graphs from different graphics procedures can all be stored in the same graphics catalog.

❸ **HAXIS/VAXIS:** The H- and VAXIS statements assign the axis definitions to the respective horizontal and vertical axis.

❹ **NAME:** A name is assigned to the graph. Explicit naming of the graphs is helpful when referencing them in the GREPLAY procedure later in the process.

Because the graphs are tailored specifically to combine them on one page there is a loss of flexibility. The graphs probably won't be usable for any other purpose. The increased quality and control has to be weighted against the potential loss of re-usability when “100%-templates” are used.

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

CREATING TITLE AND FOOTNOTE SLIDE

The GSLIDE procedure can be used to create a graph with titles and footnotes that apply to the whole page (**Figure 7**). GSLIDE converts all defined titles and footnotes into a graph. This graph can be stored like a regular SAS graph in a catalog for later use in the GREPLAY procedure.

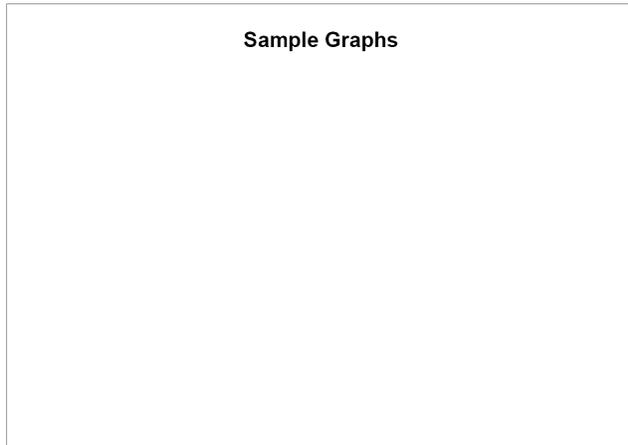


Figure 7: Example for a title slide

The following code can be used to create a title slide:

```
title1 height=1cm "Sample Graphs";

proc gslide gout❶= work.example name❷= "t_slide";
run;
quit;
```

Explanation of parameters:

- ❶ GOUT: Name of the SAS catalog where the title slide graph is stored
- ❷ NAME: Name of title slide in the SAS catalog.

Both GOUT and NAME are later used by GREPLAY procedure to reference the title slide.

Titles in a graph span the whole output area. Therefore if titles or footnotes for individual graphs are required they can best be realized using the ANNOTATE functionality.

STEP 2: CREATING “100%-TEMPLATES”

The GREPLAY procedure is used to create the so-called “100%-templates”. The characteristics of a “100%-template” is that all panels in the template cover 100% of the output area and that the panels are stacked on top of each other. For each graph in the catalog (including the title slide) that you would like to combine on one page you need to define a panel.

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”

The following code will result in 4 panels covering 100% of the output area defining a “100%-template”. Because all panels have the same dimensions the creation is packed in a macro loop.

```
proc greplay nofs① tc②= work.tempcat;
  tdef③ newtemp
    %do pos= 1 %to 4;
      &pos.④/lly⑤=0 uly=100
        llx=0 lrx=100
        lry=0 ury=100
        ulx=0 urx=100
      %end;;
run; quit;
```

Explanation:

- ① NOFS: Procedure is started in line mode. No full screen will be opened.
- ② TC: Name of catalog where the template is stored.
- ③ TDEF: This statement starts the template definition and is followed by the name of the template in the SAS catalog. In this case the name of the catalog entry is NEWTEMP.
- ④ &POS: Panel number. In this case panel numbers are 1 to 4.
- ⑤ LLY, ULY, ...: These coordinates define the output area. E.g. LLY specifies the Y coordinate of the lower left corner of the panel in percent of the output area.

Traditional template definition versus “100%-template” definition:

When a “100%-template” is defined the position of the final graph on the page is not relevant. The position of the graph is defined in the graphics procedure itself. This makes the definition of templates much easier compared to the traditional method.

The output of the GREPLAY LIST command in the log file reveals the definition of the four equally designed panels:

<u>NEWTEMP</u>		Graphics template						
Pan	Ll-x	Ll-y	Ul-x	Ul-y	Ur-x	Ur-y	Lr-x	Lr-y
1	0.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0
2	0.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0
3	0.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0
4	0.0	0.0	0.0	100.0	100.0	100.0	100.0	0.0

A traditional template can easily become complicated even for simple layouts. The following template definition represents a panel with four panels similar to the template shown in **Figure 3**:

<u>NEWTEMP</u>		Graphics template						
Pan	Ll-x	Ll-y	Ul-x	Ul-y	Ur-x	Ur-y	Lr-x	Lr-y
1	0.0	10.0	0.0	50.0	50.0	50.0	50.0	10.0
2	0.0	50.0	0.0	90.0	50.0	90.0	50.0	50.0
3	50.0	50.0	50.0	90.0	100.0	90.0	100.0	50.0
4	50.0	10.0	50.0	50.0	100.0	50.0	100.0	10.0

Pre-defined templates are available in the SASHELP.TEMPLT catalog. They can be reviewed using the PREVIEW command in the GREPLAY procedure – just in case you want to try the traditional template method.

Multiple Graphs on One Page Using GREPLAY with “100%-Templates”**STEP3: OVERLAY GRAPHS USING GREPLAY**

After the graphs and the “100%-template” have been created and stored in a SAS catalog the GREPLAY procedure can be used to combine all graphs on one page. The individual graphs are assigned to the defined panels. Because the panels are overlaid the graphs will be overlaid as well.

The following code can be used to overlay the graphs:

```
proc greplay nofs tc❶= work.tempcat template❷= newtemp gout❸= work.graphs;

    igout❹ = work.example;

    treplay❺
        1: plot1
        2: plot2
        3: plot3
        4: t_slide;

run;
quit;
```

Explanation of parameters:

- ❶ TC: Name of catalog where the templates are stored.
- ❷ TEMPLATE: Name of the template to be used by procedure.
- ❸ GOUT: Name of the catalog where the output of the GREPLAY procedure will be stored.
- ❹ IGOUT: Name of catalog, which contains the individual graphs.
- ❺ TREPLAY: Assign the individual graphs to the panels of the template. The number before the colon is the panel number followed by the name of the graph from the graphics catalog IGOUT.

CONCLUSION

Combining multiple graphs on one page using the traditional GREPLAY method is tedious and the results are often not satisfying. SAS version 9.2 should have all the tools to make this easier (the author hasn't had a chance to work with the new version). However solutions are already available in earlier SAS versions. Using stored graphs, PROC GREPLAY with 100%-templates, and overlaying graphs is a good method for creating output with multiple graphs on one page. The big advantage of this method is that scaling of axis and fonts is done when the single graph is produced. The axis lengths and font sizes will not be scaled and distorted when all graphs are combined on one page. This will result in readable axis labels and accurate display of the axis and symbols without any surprises. Mixing graphs from different procedures and with different shapes and orientations is very easy. The disadvantage is the limited re-usability of the individual graphs as their appearance is designed with the output in mind. But in the author's opinion the advantages of the method presented in this paper far outweigh the disadvantages.

REFERENCES

1. Multiple Graphs on One Page Using SAS/GRAPH V9, Hany Aboutaleb, SAS Global Forum 2008 (<http://www2.sas.com/proceedings/forum2008/222-2008.pdf>)
2. Using Just SAS 9.1 to Draw Trellis Graphs, Philip Holland, PhUSE 2006 (<http://www.lexjansen.com/phuse/2006/cc/cc02.pdf>)
3. A Programmer's Introduction to the Graphics Template Language, Jeff Cartier, SAS Global Forum 2007 (<http://www2.sas.com/proceedings/sugi31/262-31.pdf>)
4. The Rest of the Way There... Creating Multiple Graphs on a Single Page, Mary Buckwalter, Terek Peterson, PhilaSUG Fall 2007 (<http://www.philasug.org/Presentations/200711/p03.pdf>)
5. ODS and Statistical Graphics, Shi-Tao Yeh, NESUG 17, 2004 (<http://www.nesug.org/proceedings/nesug04/pr/pr02.pdf>)

ACKNOWLEDGMENTS

I would like to thank Cynthia Zender from SAS for a discussion at PharmaSUG 2008, inspiring me to materialize this paper. Thanks to Terek Peterson for his PhilaSUG material. Thanks to my colleagues Cordula Massion and Monika Kawohl at Accovion for reviewing the paper and giving me valuable feedback.

CONTACT INFORMATION

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

Dirk Spruck
Accovion
Softwarecenter 3
35037 Marburg
Germany
Work Phone: +49 6421 94849-37
Fax: +49 6421 94849-61
Email: dirk.spruck@accovion.com
Web: www.accovion.com

Brand and product names are trademarks of their respective companies.