

Presenting graphs on one page

Sandrine STEPIEN, Quintiles, Strasbourg, France

ABSTRACT

It is sometimes useful to present more than one graph on the same page. The GREPLAY and the GSLIDE procedures enable programmers to organize the graphics presentation. The GREPLAY procedure creates templates by defining panel areas (boxes) and filling them by different graphics catalog entries: graphs, titles, footnotes. Presenting more than one graph on one page should be used with caution as the dimension of graphs will be stretched and can be less than readable after re-organization.

STUDY XXXXXXXXX

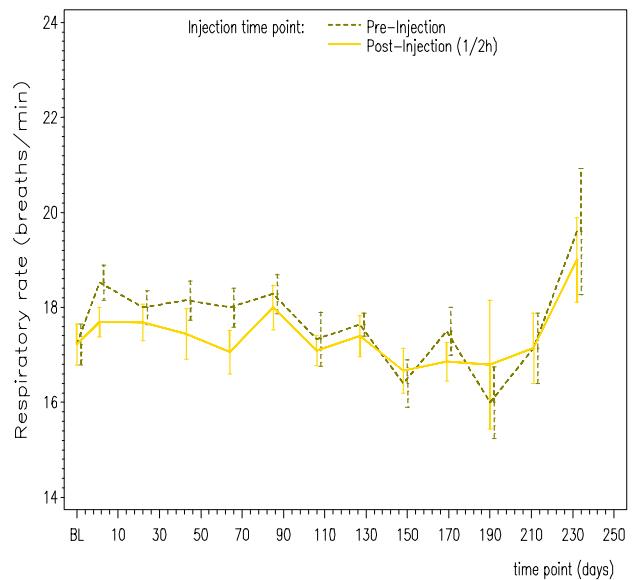
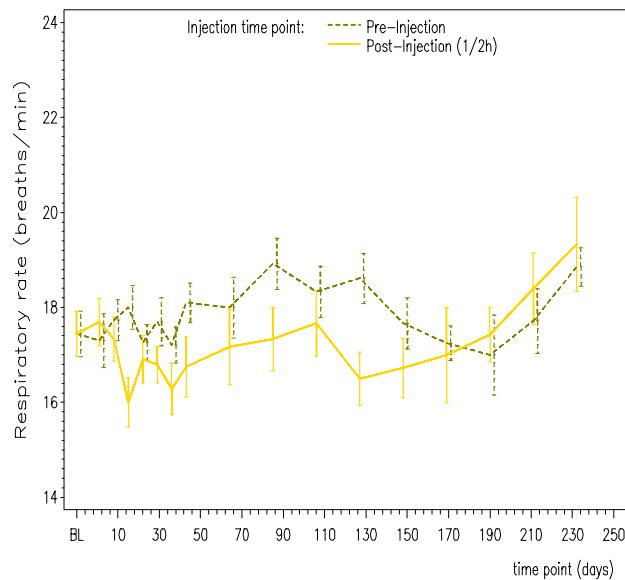
CONFIDENTIAL – Draft version

XXXXXX

Figure X : Graph of vital sign data: Mean $\pm$ SEM by time point and schedule

Schedule 1

Schedule 2



INTRODUCTION

This paper will give you some technical clues to arrange your graph presentation by creating templates and replay graphics catalog entries into those templates. By using concrete examples, we will show you how to use the GREPLAY and the GSLIDE procedures which are the two SAS® procedures that control the template parameter definition and use.

SAS® ONLINE HELP OVERVIEW

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. The GREPLAY procedure operates in both windowing and line-mode environments. With the GREPLAY procedure, you can:

- use, create, or modify templates. You can use templates to describe positioning on a single display for the graphics output that is stored in one or more catalog entries.
- manage entries in SAS® catalogs by
 - renaming, deleting, or copying catalog entries that contain graphics output, templates, and color maps
 - rearranging catalog entries that contain graphics output.
- create new graphics output by replaying one or more catalog entries into panels within a template.

(see: <http://v8doc.sas.com/sashtml/gref/z12view.htm>)

First just a reminder about the SAS/GRAPH procedures and more precisely how graphics entries are stored and named:

GRAPHICS OUTPUT

STORING GRAPHICS OUTPUT IN SAS® CATALOGS

When you run a SAS/GRAPH procedure that produces graphics output, a copy of the graphics output is always stored in a SAS® catalog. A catalog is a type of SAS® file in which you can store different types of information, called catalog entries. Catalog entries of type GRSEG store graphics output. In addition, SAS/GRAPH procedures create other types of catalog entries such as DEV, CMAP, FONT, and TEMPLATE. You can store multiple catalog entries in one catalog.

You can use catalog entries to store graphics output that you want to display again without having to rerun the program. Use the GRAPH window, the GREPLAY procedure, or the SAS® Explorer window to redisplay graphics output stored in catalog entries.

SAS® catalogs can be either temporary or permanent. Temporary catalogs are erased at the end of each SAS® session and cannot be retrieved. Permanent catalogs are retained until you delete them. Therefore, they exist after the end of a SAS® session and can be used in later SAS® sessions.

Unless you select another catalog, either temporary or permanent, SAS/GRAPH procedures always store a copy of your graph in the temporary catalog WORK.GSEG, regardless of the other forms of graphics output that you choose. By default, each graph is appended to the catalog. The graphs in WORK.GSEG are erased when you end the SAS® session.

NAMES AND DESCRIPTIONS OF CATALOG ENTRIES

SAS/GRAPH software always assigns a name and a description to each catalog entry so you can identify it. By default, the names and descriptions are determined by the procedure. For example, a graph produced by the GCHART procedure would be assigned the name GCHART and a description like PIE CHART OF MONTH. A GPLOT procedure assigns the name GPLOT.

By default, SAS/GRAPH appends each new entry to the catalog. If you create more than one graph with a procedure during a SAS® session and the graphs are stored in the same catalog, SAS/GRAPH software adds a number to the end of the name of subsequent catalog entries. This number makes the names unique within the catalog. For example, if you create three graphs with the GCHART procedure during the same SAS® session, the catalog entries are named GCHART, GCHART1, and GCHART2. SAS/GRAPH software uses this naming convention whether catalog entries are being stored in a temporary or permanent catalog.

You can supply a name and description when you create the graph by using the NAME= and DESCRIPTION= options. If you create more than one graph of the same name, SAS/GRAPH software increments the specified name just as it does the default names.

PROC *procedure-name* **GOUT**=<libref.> *catalog-name*

<i>procedure-name</i>	specifies the graphics procedure you are running
<libref.> <i>catalog-name</i>	specifies the name of a catalog where you want the output stored. If the specified catalog already exists, the procedure simply adds a catalog entry to the catalog. If the catalog does not exist, it is created

For the other examples described in this paper, we will consider that the following graphs are stored in a temporary catalog <work.GRAPHOUT> : GPLOT, GPLOT1, GPLOT2 and GPLOT3.

CREATE TEMPLATES/GREPLAY PROCEDURE

SYNTAX

To define templates the following syntax is required:

```
PROC GREPLAY          TC=SAScatalog nofs;
  TDEF TPL_name DES='entry-description blablablabla'

    Panel-number1/      llx=val1      lly=val1
                        ulx=val1      uly=val2
                        urx=val2      ury=val2
                        lrx=val2      lry=val1

    Panel-number2/      ... ;

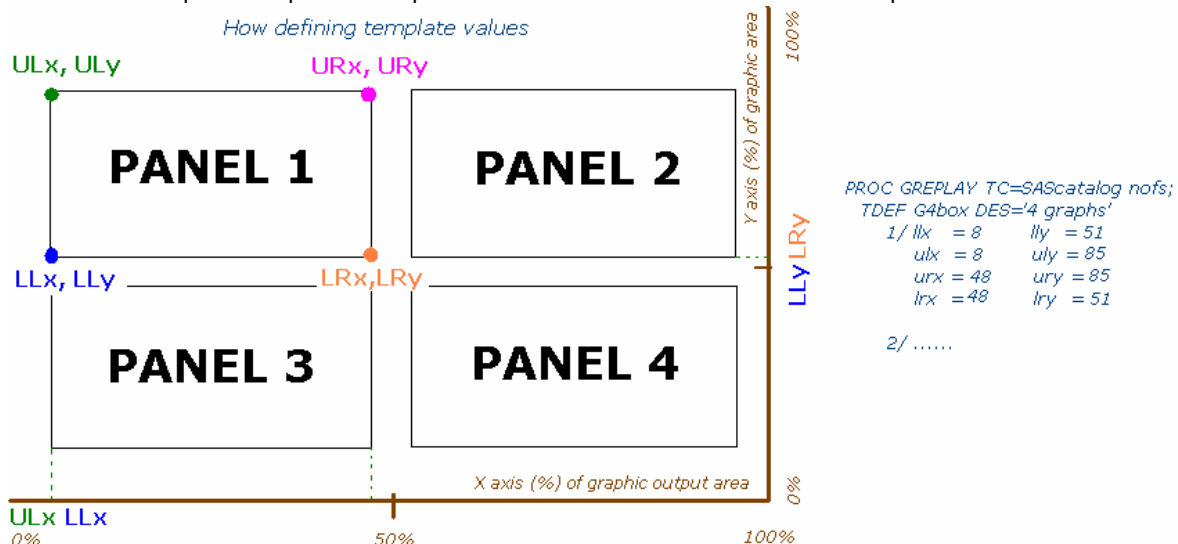
run;
quit;
```

TC	identifies the SAS® catalog where templates are to be stored or identifies the name of a SAS® catalog that contains templates.
Nofs	specifies that the GREPLAY procedure should use line mode.
TDEF	identifies an existing or a new template. If the template is not in the template catalog, the procedure creates it. If the template is already in the template catalog, the procedure modifies or makes additions to that template.
DES	specifies the description of the catalog entry for the template
Panel number	identifies the number of the panel that is being defined or modified
Panel corner coordinates:	LOWER LEFT: llx/lly UPPER LEFT: ulx/uly UPPER RIGHT: urx/ury LOWER RIGHT: lrx/lry specifies the (X,Y) coordinates of the panel corners. Units for x and y are percentage of the graphics output area

DEFINING PANELS

Panels are box areas in which graphics output will be loaded and will fit the panel area. Panels are defined using panel corner coordinates: upper and lower, right and left corners with X and Y axis as a percentage of the graphic output area (page). Each panel is identified by a panel number with panel corner coordinates.

The following picture shows how to organize the template and how to define the values corresponding to each panel corner coordinates. This example is a 4 panel's template and we focus on the first one for description:



PhUSE 2006

To have more details about the syntax of the GREPLAY with all options and statements, please refer to the SAS® online doc V8 :

<http://v8doc.sas.com/bin/ixcgisol/sashtml/gref/z12ntax.htm?query=%22proc%22+AND+%22greplay%22>,
<http://v8doc.sas.com/sashtml/gref/z18ment.htm>, and <http://v8doc.sas.com/sashtml/gref/z19ment.htm>.

In the panel 1 area, the graphics output will be loaded and will be stretched to fit the whole area. So when defining the panel area, the proportion of the original graph should be kept. As the graph height and width will be reduced, you should ensure that the graph remains readable (axis definition and values...)

EXAMPLE1:

The following code will store the template named _Test1 in the SAS® catalog named work.tempgraf. This template is defined by 4 panels and described as “4 graphs – test1”:

```
PROC GREPLAY TC=work.tempgraf NOFS;
  TDEF G4_Test1 DES='4 graphs - test1'
    1/l1x=8      l1y=45
      ulx=8      uly=85
      urx=48     ury=85
      lrx=48     lry=45

    2/l1x=52     l1y=45
      ulx=52     uly=85
      urx=92     ury=85
      lrx=92     lry=45

    3/l1x=8      l1y=2
      ulx=8      uly=42
      urx=48     ury=42
      lrx=48     lry=2

    4/l1x=52     l1y=2
      ulx=52     uly=42
      urx=92     ury=42
      lrx=92     lry=2
quit;
run;
```

TEMPLATE VIEW

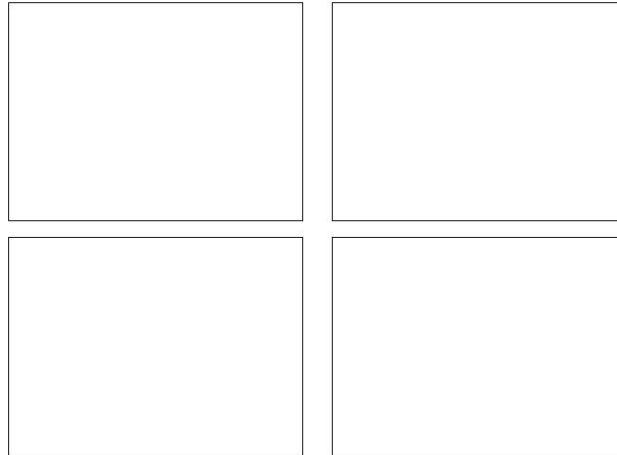
Some PROC GREPLAY statements are used to have an overview of template definitions stored in specific catalogs. The syntax below is used to put in the log window the values of each panel corner coordinate and to preview the graphics template.

Syntax

```
PROC GREPLAY TC=work.tempgraf NOFS;
  TEMPLATE g4_test1;
  LIST template;
  PREVIEW g4_test1;
quit;
run;
```

LOG window:

G4_TEST1	4 graphs - test1									
Pan	Clp	Color	L1-x	L1-y	U1-x	U1-y	Ur-x	Ur-y	Lr-x	Lr-y
1			8.0	45.0	8.0	85.0	48.0	85.0	48.0	45.0
2			52.0	45.0	52.0	85.0	92.0	85.0	92.0	45.0
3			8.0	2.0	8.0	42.0	48.0	42.0	48.0	2.0
4			52.0	2.0	52.0	42.0	92.0	42.0	92.0	2.0

Preview SAS/GRAPH window**TITLE GRAPHICS OUTPUT****CREATE TITLE GRAPHICS OUTPUT**

A global graphics title/footnote is required most of the time to make your final graphics output clearer. To create an overall title/footnote, the usual method cannot be used. A title/footnote graphics output entry needs to be generated. To obtain that graphics entry, the use of PROC GSLIDE is required. The syntax is the following:

```
PROC GSLIDE GOUT=graf NAME='titre';
  TITLE1 j=c h=2.5 "Protocol YYYY";
  TITLE2 j=c h=2.5 "CONFIDENTIAL - draft version - &sysdate9";
  TITLE4 j=c h=2.5 "Figure X : Blablablablabla";
  FOOTNOTE j=c h=2.5 "Page 1 of 1";
run;
quit;
```

GOUT	specifies the SAS® catalog in which to save the graphics output produced by the GSLIDE procedure. If you omit the libref, SAS/GRAPH looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist
NAME	specifies the name of the catalog entry for the graph. The maximum length for entry-name is eight characters. The default name is GSLIDE. If the specified name duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry, for example, GSLIDE1

For more details about the GSLIDE procedure please refer to the SAS® online doc:

[http://v8doc.sas.com/bin/ixcgisol/sashtml/gref/z0212137.htm?query=%22proc%22+AND+%22slide%](http://v8doc.sas.com/bin/ixcgisol/sashtml/gref/z0212137.htm?query=%22proc%22+AND+%22slide%22)

This leads to the next part : how to define the panel for the overall title?

TITLE PANEL IN TEMPLATE DEFINITION

The dimension of graphics output, created by the PROC GSLIDE, refers to a page area: title part is written on the top of this area and footnote part at the bottom. But, if e.g. no footnote is defined, the resulting output keeps the same page dimension. In those conditions, the corner coordinates defined for the title panel should have the coordinates of an entire page area (see table below). If it was not the case, the title graphics output would be reduced to fit the panel dimensions. So height option, defined in the TITLE statement of PROC GSLIDE, would not be respected.

The DEF option of TDEF statement is used to define the title panel which has the following characteristics:

Default panel corner coordinates values	
Panel corner	coordinates
Lower left	(0,0)
Upper left	(0,100)
Lower right	(100,100)
Upper right	(100,0)

So to define the title panel area, the syntax is described below:

```
PROC GREPLAY TC=work.tempgraf NOFS;
      TDEF G4_Test2 DES='1 title and 4 graphs - test2'
1/llx=8      lly=45
   ulx=8      uly=85
   urx=48     ury=85
   lrx=48     lry=45

      2/...   3/...   4/...   (see example 1 for definitions)

      5/DEF;

quit;
run;
```

REPLAYING ENTRIES INTO A TEMPLATE

To organize the new graphics output using the template defined in example 1, each graphics entry has to be assigned to a pre-defined panel by using the following syntax:

```
PROC GREPLAY IGOUT=work.graphout GOUT=work.TempOut TC=work.tempgraf nofs;

      TEMPLATE= G4_Test2;
      TREPLAY      1:gplot
                   2:gplot1
                   3:gplot2
                   4:gplot3
                   5:titre;

run;
quit;
```

where gplot, gplot1, gplot2, gplot3 are graphics outputs generated by a PROC GPLOT. Those graphics entries are stored in the catalog named *work.GraphOut*. The title graphics output generated by the PROC GSLIDE is stored in the same catalog and named *Titre*. The resulting template graphics output is stored in the catalog named *work.TempOut*.

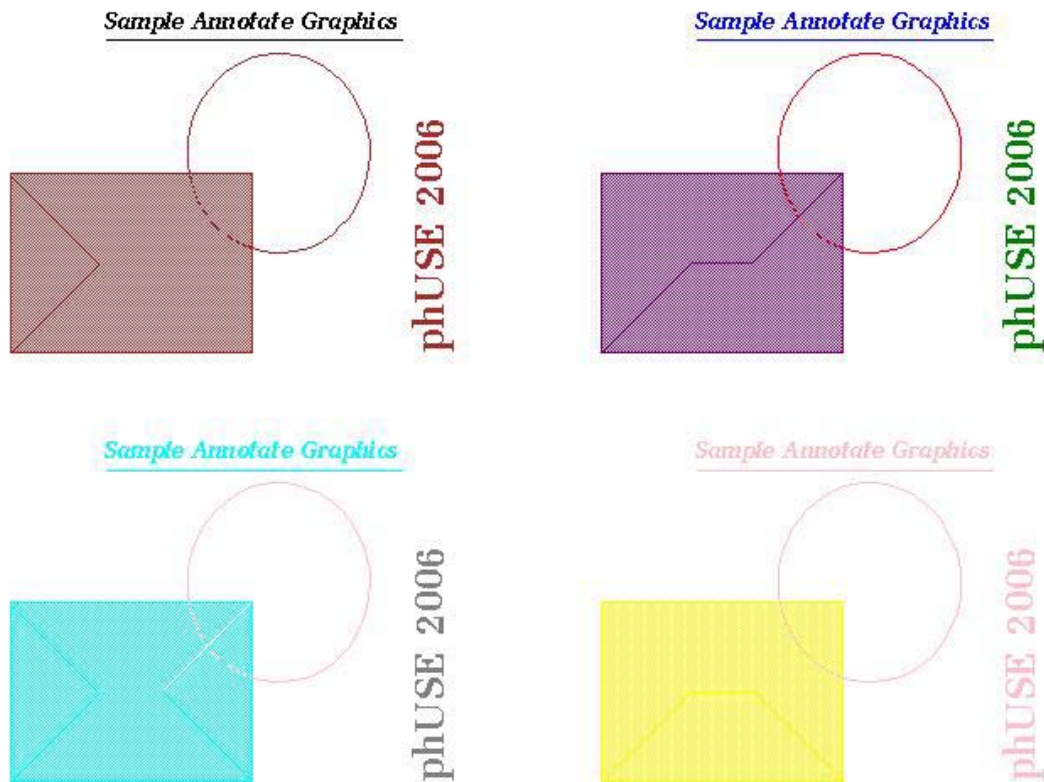
IGOUT	specifies the input catalog to use with the GREPLAY procedure. The input catalog that you specify with the IGOUT= option should be a catalog that contains the graphics output that will be replayed. If you omit the libref, SAS/GRAPH looks for the catalog in the temporary library called WORK. Input-catalog can be the same catalog that you specified in the GOUT= option
GOUT	specifies the SAS® catalog in which to save the graphics output that is produced by the GREPLAY procedure. In addition, catalog entries that contain graphics output can be copied to output-catalog . If you omit the libref, SAS/GRAPH looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist. Output-catalog can be the same catalog that is specified in the IGOUT= option.

PhUSE 2006

TEMPLATE	assigns a current template to use when replaying graphics output where template-entry names an existing template in the template catalog that is specified in the TC= option. If template-entry is not in the catalog, an error message is written to the SAS® log. Template-entry must have a catalog entry type of TEMPLATE
TREPLAY	Selects one or more catalog entries from the same catalog for replay in template panels

Protocol YYYYY

CONFIDENTIAL - draft version - 01SEP2006
Figure X : Blablablabla



Page 1 of 1

CONCLUSION

The syntax for defining a template might appear quite complex at first, but once this has been used a couple of times, it becomes easier. The GREPLAY and GSLIDE procedures need to be manipulated with caution as proportion of graphics entries could be changed by defining inadequate panel area. The programmer will also need to check if graphs remain readable after incorporating them into the template.

Furthermore, for some reasons linked to the automatic graph fitting, it is sometimes required to create one box and fill it with one graph to obtain the awaited presentation result (e.g.: entire frame).

REFERENCES

SAS® online doc V8

CONTACT INFORMATION (HEADER 1)

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

Sandrine STEPIEN

QUINTILES

4 route de la rivière

67380 LINGOLSHEIM

Work Phone: +33(0)3 88 77 44 13

Email: Sandrine.stepien@quintiles.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.