

Paper 159-31

ODS LAYOUT is Like an Onion

Rich Mays, University of Rochester Medical Center, Rochester, NY

ABSTRACT

ODS LAYOUT is like an onion. They both make you cry? No! They both have layers!

In version 9.1, the ODS LAYOUT feature is one of the best inventions since the Carmel Frappuccino. SAS® now gives you a straightforward approach to placing graphs and tables on a page while eliminating the need for PROC GREPLAY and TEMPLATES. This paper demonstrates a method of using ODS LAYOUT capabilities to create PDF documents by creating a series of page “layers.” It also demonstrates the use of macros to enhance the capabilities of ODS LAYOUT. A macro controlled system to determine available space on a page, and the subsequent placement of graphs/tables will be discussed. A good understanding of how to use macros is essential to understanding this paper. An understanding of ODS LAYOUT and PROC GREPLAY is helpful but not necessary. This paper will include actual code and examples.

BACKGROUND

The Office of Clinical Practice Evaluation at the University of Rochester Medical Center creates a number of PDF documents utilizing SAS® which are published to our departmental web site. Over the last couple of years, we have been developing techniques to enhance our reporting. We have been producing reports in PDF format that present multiple graphs and tables on larger sheets of paper (11 by 17) for use as posters, color backgrounds to make the individual pages easier to read, and color borders to highlight key graphs. In making the individual graphs stand out, we wanted to be able to easily add these features without creating a lot of complicated code that would be difficult to maintain. In addition, any solution needed to be capable of creating both single and multiple page PDF files. In the past we used GREPLAY with TEMPLATE definitions to create such pages. This resulted in code that was difficult to maintain, especially if the work was transitioned to someone without solid macro experience. Fortunately, ODS LAYOUT came to the rescue!

In this presentation, I will show how I went about creating PDF files using the ODS LAYOUT feature of SAS® 9.1. All of the reports in this presentation were created on a Unix server. I would expect these programs to work on other platforms.

INTRODUCTION

Some assumptions for this paper –

1. There are several ways to accomplish the desired output in this presentation. I chose methods that I am most familiar with and felt could be presented most clearly.
2. The objective of this paper is to demonstrate how we created a report through a series of generated “layers” and to compare ODS Layout to an alternative method
3. Simpler code is used for presentation purposes. When using more complicated code, it is even more important to have a simpler method.
4. The graphs generated for this presentation came from the ORSALES dataset in the SASHELP library. Summary data used to create the final graphs were taken from the SASHELP.ORSALES dataset and were summarized using PROC SQL.

ODS LAYOUT – WHAT IS IT?

ODS LAYOUT is a new feature in SAS® version 9. Using it allows you to create a PDF file on the fly. Using ODS LAYOUT, you can assemble a report by placing a series of page “layers” on the page. A useful advantage of doing this is that you don’t have to worry about background colors from the previous layer showing up through the current layer’s individual graphs. It’s the equivalent to matting a photograph on top of a color sheet of paper – the border (mat) enhances and draws attention to the picture. Any colors from the individual graph cover up the colors from the previous layer. A similar report can be generated using GREPLAY with annotation. However, ODS LAYOUT offers more control (for example, using inches instead of pct for graphics units) than GREPLAY with annotation.

The result of using ODS LAYOUT?

- PDF pages can be assembled in a more straightforward manner. Adding background colors and graph borders is relatively simple
- Your program may be a little longer but easier to follow
- PDF pages can include output from graphics AND other reporting procedures
- Each PDF page can have a different layout with a varying number of graphs per page without a lot of complicated code.

CREATING A PDF DOCUMENT USING ODS LAYOUT

To create a PDF document using ODS LAYOUT, there are a few statements you need to use. You will need to tell SAS® about the PDF file itself (name, location, etc.), the starting and ending points for each page, what output is placed on the page, and where to position the output.

The [ODS PDF](#) and [ODS PDF CLOSE](#) statements are used to generate the PDF file. In general, these statements will be used to tell SAS® what you want to call the file and where you want to store it.

The [ODS LAYOUT START](#) and [ODS LAYOUT END](#) statements are used to generate one page for a PDF document. You will place the ODS LAYOUT START statement at the start of a block of code for a given page and the ODS LAYOUT END at the end of the block. We will use the basic statements for this presentation (no options).

The [ODS REGION](#) statement allows you to specify where on a page you would like to place output from a graphics/report procedure.

SAS® expects the first ODS REGION statement to immediately follow the ODS LAYOUT START statement. The ODS REGION statement allows flexibility in how you specify the location for graph/report output – you can base placement on percentage of the page, inches, and centimeters, just to name a few. Each of these measurements is understood as the distance from the top left corner of the page. In contrast, PROC GREPLAY and annotate datasets use the lower left corner as a starting point. For most people, the top left corner is the most intuitive.

Since the GREPLAY procedure uses percentages to create page templates, we will use percentages for the ODS REGION statement so that we can more easily compare the two methods presented.

REPORT REQUIREMENTS

For this presentation, I will create a PDF document with the following characteristics:

1. Each page will show multiple graphs. Each graph will be the same size and shape.
2. Each page will have a color background. In this case we will use cyan. There is the option to use a background image but we will stick with the simpler background.
3. Individual graphs will have a white background and will have a color border. This means that the background for the entire individual graph will be white, not just the plotting area.

```
ODS PDF file="<file name>" <options>;
ODS LAYOUT START <options>;
ODS REGION <graph location & size>
  SAS Graphics/Report Code
ODS REGION <graph location & size>
  SAS Graphics/Report Code
ODS REGION <graph location & size>
  SAS Graphics/Report Code
ODS LAYOUT END ;
ODS PDF CLOSE ;
```

METHODS FOR CREATING THE PDF DOCUMENT

For demonstration purposes, I will show two different methods for creating the overall report. Each method will use a series of GPLOTS and GSLIDES to create the individual graphs, backgrounds, and borders necessary for the overall report.

ODS LAYOUT METHOD

- ODS Layout used to set up page layouts as each page is created
- ODS Layout used to create page background and border colors
- Individual graphs will be created as the page is being assembled, although creating them ahead of time is also an option

ALTERNATIVE METHOD (GREPLAY)

- Will use a combination of PROC GREPLAY with annotation from a dataset.
- PROC GREPLAY will be used to set up page layouts
- An annotation dataset will include statements to create the page background and border colors. PROC GSLIDE will be called with the ANNO option to generate the background/border colors.
- All individual graphs/slides will be created first and then replayed into the PDF document.
- PROC GREPLAY will be used to create the final PDF document

CREATING THE BACKGROUND COLOR

How do we put the background color on the page? Two methods for doing so create essentially the same result. There are subtle differences between them, but not enough to worry about.

ODS LAYOUT METHOD

To create this background using ODS LAYOUT, we need to tell SAS® how large our “graph” will be (in this case we will use 100% of the page), specify the page background color, and call a GSLIDE statement.

```
ods region x=0 pct
          y=0 pct
          height=100 pct
          width =100 pct;

options cback=cyan;
title1 h=0.5 in "SUGI 31";

proc gslide;
run;
quit;
```

GREPLAY METHOD

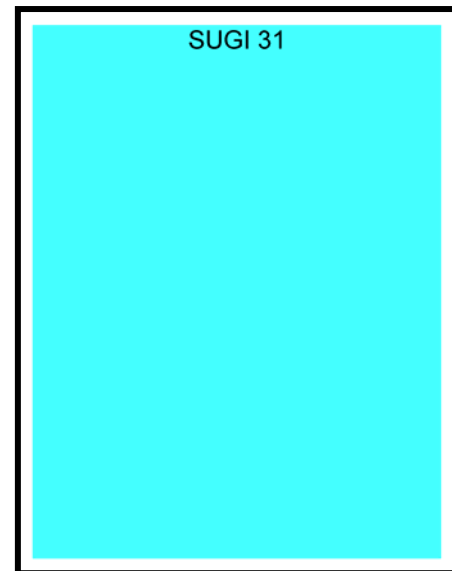
To create this background using the GREPLAY method, we will annotate the color onto the page using an annotation dataset with a GSLIDE statement. The annotation dataset will color the background by telling SAS® to draw a “bar” which is the size of the page.

The DCLANNO macro automatically sets the length and data type for all annotate variables except the text variable. The SYSTEM macro defines the annotate reference system.

```
data background;
  %dclanno;
  %system(3,3,3);
  %bar(0,0,100,100,cyan,1,s);
run;

title1 h=0.50 in "SUGI 31";

proc gslide gout=graphs anno=background;
run;
quit;
```



CREATING THE INDIVIDUAL GRAPH COLORS

How do we put color borders around the graphs? One method is to utilize the GREPLAY procedure and specify a color border. However, this does not give us much control over the size of the border. If we want that control, it is best to place rectangles "underneath" the graphs which will appear as borders when we are finished.

ODS LAYOUT METHOD

To create borders using ODS Layout, we will tell SAS® to place rectangles in the correct locations on the page. To control the width of the border lines, we will make the rectangle slightly larger than the graph area and slightly offset. The next step will be to place the graphs "layer" on top of the rectangles, eliminating the rectangle "fill." Unlike the GREPLAY method, the underlying color does not show through the graph.

```
ods region x=0 pct   height=100 pct
          y=0 pct   width =100 pct;

goptions cback=cyan;
title1 h=0.50 in "SUGI 31";

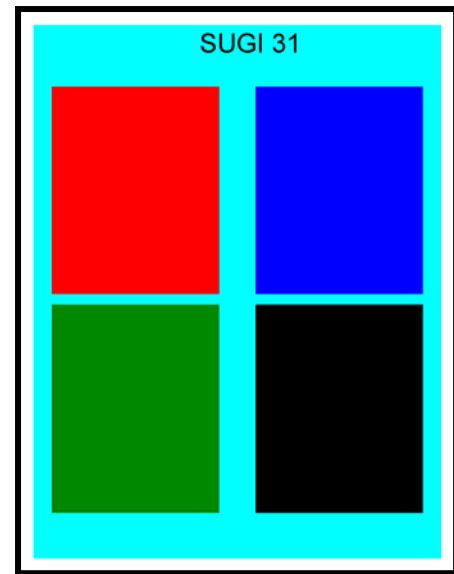
proc gslide;
run;
quit;

ods region x=4.5 pct   height=39 pct
          y=11.5 pct   width =41 pct;
goptions cback=red;
proc gslide;
run;
quit;

ods region x=54.5 pct  height=39 pct
          y=11.5 pct  width =41 pct;
goptions cback=blue;
proc gslide;
run;
quit;

ods region x=4.5 pct   height=39 pct
          y=52.5 pct   width =41 pct;
goptions cback=green;
proc gslide;
run;
quit;

ods region x=54.5 pct  height=39 pct
          y=52.5 pct  width =41 pct;
goptions cback=black;
proc gslide;
run;
quit;
```



GREPLAY METHOD

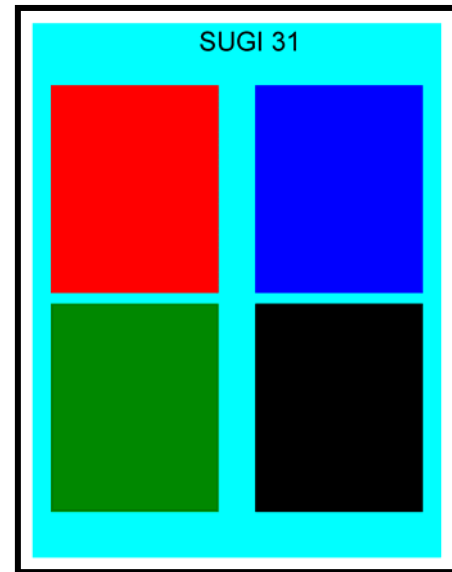
Creating borders with PROC GREPLAY involves one more step than ODS LAYOUT. We will annotate the rectangles onto the page over the background layer, slightly larger than the final individual graph areas. However, graphs placed on top of the rectangles do not “cover” the border color. [The background color shows through.](#) How do we ensure that the graph is on a white background? We need to “remove” the rectangle fill by overlaying the color rectangles with slightly offset white ones.

To add the rectangles, we just need to call the graphics macro, bar, to annotate the color onto the page:

```
data background;
  %dclanno;
  %system(3,3,3);
  %bar(0,0,100,100,cyan,1,s);

  %bar(04.5,49.5,45.5,88.5,red,1,s);
  %bar(54.5,49.5,95.5,88.5,blue,1,s);
  %bar(04.5,07.5,45.5,47.5,green,1,s);
  %bar(54.5,07.5,95.5,47.5,black,1,s);
run;

title1 h=0.5 in "SUGI 31";
proc gslide gout=graphs anno=background;
run;
quit;
```



To “remove” the rectangle fill, we place white rectangles onto the color ones. The size of the border line is dependent on how the size and offset of the white rectangle as it overlays the color rectangle. The GSLIDE code will remain the same. Only the “background_color” dataset changes.

To make this easier, the example below contains a variable, width, to represent the border line width. SAS will automatically offset and size the white rectangle using that variable.

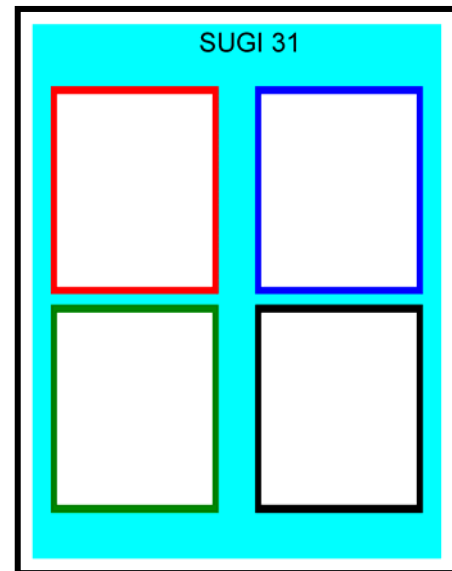
```
data background_color;
  retain width 0.5; /* Border Width */

  %dclanno;
  %system(3,3,3);

  %bar(0,0,100,100,cyan,1,s);

  %bar(05-width,08-width,
    45+width,47+width,green,1,s);
  %bar(05-width,50-width,
    45+width,88+width,red,1,s);
  %bar(55-width,08-width,
    95+width,47+width,black,1,s);
  %bar(55-width,50-width,
    95+width,88+width,blue,1,s);

  %bar(05,08,45,47,white,1,s);
  %bar(05,50,45,88,white,1,s);
  %bar(55,08,95,47,white,1,s);
  %bar(55,50,95,88,white,1,s);
run;
```



Offset Example

CREATING THE FULL PAGE

HELPFUL MACROS

In order to create the final page using ODS LAYOUT, we need to repeat a similar block of code to create the border "rectangle" color and to plot the individual graphs. To make this task a little easier to present rather than showing several pages of code, I have created a macro to allow SAS® to create the border and plot the graphs for each individual graph.

The GREPLAY method will also create rectangles/borders and plot the individual graphs. However, because of the structure of the code, only one of the macros, create_graph, will be necessary to call.

The first macro, [create_graph](#), is used to create a single plotted graph. It accepts one parameter, category, which is plotted when the macro is called. It is used by the ODS LAYOUT and the GREPLAY methods.

```
%macro create_graph (category=);
  title h=0.5 in "&category Quarterly Profit";
  proc gplot data=quarterly_profit_averages gout=graphs;
    where Product_Category="&category";
    plot mean_profit*quarter / haxis=axis1;
  run;
  quit;
  title;
%mend;
```

The second macro, [create_border](#), is used to create a border around an individual graph by creating a color rectangle and placing the graph on "top" of it. This macro is used by the ODS LAYOUT method only.

The parameters used are as follows:

- x = The horizontal starting point for the rectangle and is the distance from the page's left edge.
- y = The vertical starting point for the rectangle and is the distance from the page's top edge.
- hgt = The rectangle height.
- width = The rectangle width
- border_color = The rectangle color
- border_width = The border line thickness after the graph is placed over the rectangle.
- category = The category that the will be plotted (used to call [create_graph](#) macro)

```
%macro create_border(x=,y=,hgt=,width=,border_color=black,border_width=0.5,category=);
*-----*
  Part 1 - Create the color rectangle for the border
*-----*
  ODS REGION x=&x pct      height=&hgt pct
             y=&y pct      width =&width pct;
  goptions cback=&border_color;
  proc gslide;
  run;
  quit;

*-----*
  Part 2 - Adjust x, y, height, width values based on border_width
*-----*
  %let x      = %sysevalf(&x + &border_width);
  %let y      = %sysevalf(&y + &border_width);
  %let hgt    = %sysevalf(&hgt - 2*&border_width);
  %let width  = %sysevalf(&width - 2*&border_width);

*-----*
  Part 3 - Create the color rectangle for the border
*-----*
  ods region x=&x pct      height=&hgt pct
             y=&y pct      width =&width pct;
  goptions cback=white;
  %create_graph (category=&category)
%MEND;
```

ODS LAYOUT METHOD:

To complete the full page using ODS LAYOUT, we need to add code to call the `create_border` macro with the appropriate location, size, border, and category information. As mentioned earlier, the `create_border` macro is simply placing the appropriate rectangle color onto the page and overlaying that with the corresponding plotted graph. Note – the code below includes the NOTOC option in the ODS PDF statement. This is used so that the final PDF document does not include bookmarks.

```

symbol1 color      = blue
      interpol      = join
      value         = star
      width         = 8;
axis1  split       = 'Q';

options nonumber nodate;
title;

ods listing close;
ods pdf file="ods_layout_page.pdf" notoc;

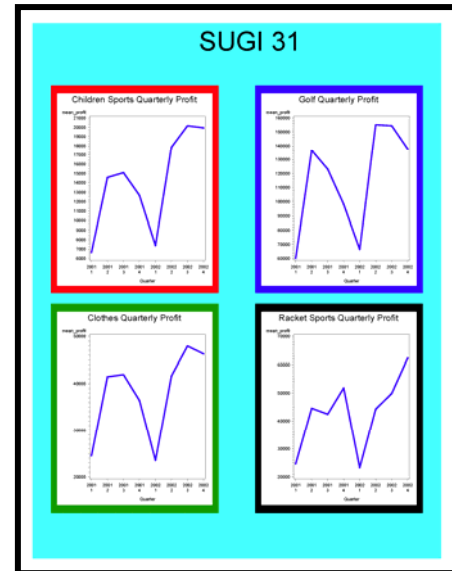
ods layout start;
ods region x=0 pct height=100 pct
      y=0 pct width =100 pct;
goptions cback=cyan;
title1 h=0.5 in "SUGI 31";
proc gslide;
run;
quit;

title;

%create_border(x=5, y=12, hgt=38, width=40, border_color=red,
      category=Children Sports)
%create_border(x=55, y=12, hgt=38, width=40, border_color=blue,
      category=Golf)
%create_border(x=5, y=53, hgt=38, width=40, border_color=green,
      category=Clothes)
%create_border(x=55, y=53, hgt=38, width=40, border_color=black,
      category=Racket Sports)

ods layout end;
ods pdf close;

```



GREPLAY METHOD

The final page will require a different set up for the GREPLAY method than with ODS Layout. To make the code cleaner, I have included some SAS provided macros.

Below, the TDEF statement describes the page layout (see example for a visual representation of the first two sections). Each page section is represented by a section number and four pairs of coordinate codes (one for each corner). Each three letter code indicates an **U**pper or **L**ower, **L**eft or **R**ight, and horizontal (**X**) or vertical (**Y**) coordinate for a section corner. For example, the **ULX** code represents the horizontal (X) coordinate value for the section's upper left corner.

```
data background_color;
  retain width 0.5; /* Border width */
  %dclanno;
  %system(3,3,3);

  %bar(0,0,100,100,cyan,1,s);
  %bar(05-width,50-width,45+width,88+width,red ,1,s);
  %bar(55-width,50-width,95+width,88+width,blue ,1,s);
  %bar(05-width,08-width,45+width,47+width,green,1,s);
  %bar(55-width,08-width,95+width,47+width,black,1,s);
  %bar(05,08,45,47,white,1,s);
  %bar(05,50,45,88,white,1,s);
  %bar(55,08,95,47,white,1,s);
  %bar(55,50,95,88,white,1,s);
run;

title1 h=0.5 in "SUGI 31";

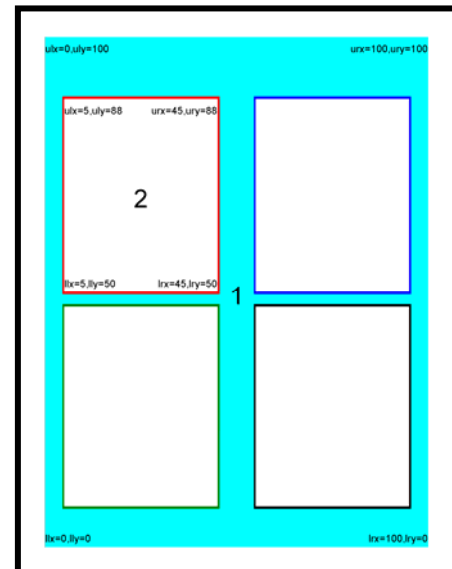
proc gslide gout=graphs anno=background_color;
run; quit;

%create_graph(category=Children Sports)
%create_graph(category=Golf)
%create_graph(category=Clothes)
%create_graph(category=Racket Sports)

filename output 'greplay_page.pdf';

proc greplay nofs tc=tempcat igout=graphs;
  tdef square4
    1 / ulx=0 uly=100 urx=100 ury=100
      llx=0 lly=0 lrx=100 lry=0
    2 / ulx=05 uly=88 urx=45 ury=88
      llx=05 lly=50 lrx=45 lry=50
    3 / ulx=55 uly=88 urx=95 ury=88
      llx=55 lly=50 lrx=95 lry=50
    4 / ulx=05 uly=47 urx=45 ury=47
      lrx=45 lry=08 llx=05 lly=08
    5 / ulx=55 uly=47 urx=95 ury=47
      llx=55 lly=08 lrx=95 lry=08;

  template square4;
  treplay 1:1
          2:2
          3:3
          4:4
          5:5;
run; quit;
```



TDEF Example

CONCLUSION

Confused yet? Hopefully not!

Both methods presented work to create our PDF files.

ODS LAYOUT is a feature in version 9 which greatly improves how PDF files can be generated. ODS Layout code is more flexible/straightforward than the code for GREPLAY. It is also easier to maintain. Finally, ODS Layout allows you to include output from non-graphics procedures, something Proc GREPLAY cannot do.

The values presented in this paper were calculated by hand so that exact coordinates could be displayed. I have written macros to automate the process of calculating the coordinates and placing the graphs on the page (based on number of columns, number of rows, and spacing parameters). Please feel free to contact me if you would like a copy of the macros.

The “tricky” part in using ODS LAYOUT is specifying to SAS® how each page is set up and where to place individual graphs. You need to supply SAS® with exact coordinates and output size for each graph produced. I don't like to spend my time doing calculations for coordinates and graph sizes. I prefer to have SAS® do the work for me whenever I can. For that reason, I wrote a couple of macros to help me out.

One macro determines the sizes of the individual graphs based on the number of columns/rows on a page, along with some other key values. It creates several global macro variables that are necessary for the additional macro.

The additional macro will tell SAS® where to place your output on the page based on the column and row you provide. It will also place a border around your graph if you specify a border color.

These two macros were not included as part of this presentation. However, if you are interested in them, please provide me with your e-mail address and I'll be happy to send you the macro code.

TRADEMARK CITATION

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.

CONTACT INFORMATION

If you would like to get a copy of any of the code used in this presentation or have any feedback related to any aspects of this paper, you may contact me at:

Rich Mays
University of Rochester
1325 Mt. Hope Ave., Suite 101
Rochester, NY 14620
E-mail: Rich_Mays@urmc.rochester.edu.