Use of Advanced ODS Features to Generate Multiple Outputs on a Single Page

Niraj J. Pandya, eClinical Solutions, A Division of Eliassen Group

ABSTRACT

Generation of multiple graphs or tables on a single page might sound a difficult task with SAS® 8 and older versions but it has become an easy task now with advanced features of SAS 9. Generating such outputs were demanding great set of programming skills in the past. But now this task is as easy as 1 2 3. Such outputs are generally a multiple frames one window kind of reports and it is hard to get such outputs through conventional SAS programming which usually gives listing form of output. Advance features of Output Delivery System (ODS) are greatly helpful to overcome the limitations of traditional SAS output. This paper is focusing on various features of ODS, new powerful SAS Version 9 capabilities and functionalities of output destinations like HTML and PDF to get multiple graphs, charts, maps, listings and tables on a single page.

Use of ODS tagsets to create HTML panels and by that generating a tabular and graphical output together on a single page with ActiveX functionality of SAS is described in detail in this paper. New ODS LAYOUT capability of SAS is also explained which can insert multiple tabular and graphical forms of reports in a single PDF page. This paper is intended for intermediate level of SAS users working on Windows/UNIX platforms.

INTRODUCTION

As SAS advances with newer versions, more and more features are becoming handy and so expectations from the programmer are also getting bigger day by day. Tasks that could not be achieved in the past are now achievable and tasks that were demanding great set of SAS programming skills are now easy with SAS V9 and higher. One of such tasks is generating multiple outputs on a single page. Generating tabular and graphical outputs, that too of different sizes on a same page is not straight forward even with the new features available and so it is in the advantage of programmer to be familiar with the new features and have good knowledge of latest development regarding these features.

As always, there are multiple ways to accomplish the same task in SAS, there are couples of methods to get multiple graphical and tabular outputs on one single page too. In the past we have used GREPLAY with the use of TEMPLATE definitions to create such reports, but it always results in code enough complex to maintain and job becomes tougher if transition has to be occurred.

But since first introduced in version 7 of SAS ®, Output delivery system (ODS) has come to rescue for such complex task. Output delivery system is designed to overcome the limitations of traditional SAS output. Traditional SAS output is in listing format by default. But ODS is flexible and can be used to create custom layouts and reports. In the newer version of SAS (v9.0 and onwards), ODS TAGSETS and ODS LAYOUT with the use of HTML and PDF destination has been real boon when it comes to generate multiple graphical and tabular reports on one page. In this presentation, we will go through these methods to get multiple reports on a single page.
DATA

For illustration and presentation purpose, I have made up following Book Sales Data. We will use this data as reference in every method presented in this paper.

**Book Sales Dataset:**

<table>
<thead>
<tr>
<th>Country</th>
<th>Region</th>
<th>Month</th>
<th>Quarter</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USA</td>
<td>North</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>USA</td>
<td>North</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>USA</td>
<td>North</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>USA</td>
<td>North</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>USA</td>
<td>North</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>USA</td>
<td>North</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>USA</td>
<td>North</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>USA</td>
<td>North</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>USA</td>
<td>North</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>USA</td>
<td>North</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>USA</td>
<td>North</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>USA</td>
<td>North</td>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>USA</td>
<td>South</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>USA</td>
<td>South</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>USA</td>
<td>South</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>USA</td>
<td>South</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>USA</td>
<td>South</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>USA</td>
<td>South</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>USA</td>
<td>South</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>USA</td>
<td>South</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>USA</td>
<td>South</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>22</td>
<td>USA</td>
<td>South</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>USA</td>
<td>South</td>
<td>11</td>
<td>4</td>
</tr>
<tr>
<td>24</td>
<td>USA</td>
<td>South</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

Sales values are in $ amount.

**ODS TAGSETS**

Use of ODS tagsets feature in SAS V9.1 is a great help for generating multiple reports in a single page. With the use of ActiveX device and Java functionality of SAS V9.1, it is easy to generate HTML output which can be internet ready output. With this method, multiple tabular and graphical outputs can be forced and set to different panels on a single HTML page. Once HTML page is created from SAS outputs, you can modify any of the output with the use of ActiveX and Java features without even modifying SAS code.

To start with, you need to divide a single page in different HTML panels. Then each panel has to be divided in number of cells by rows and columns from top left to bottom right direction. You can divide each page in multiple panels. Then you can divide each panel in multiple rows and each row can be divided in multiple cells by multiple columns in it. The beauty of this feature is that you can divide a single page in as many cells as needed based on the requirements.
In the figure below, a page is divided in two panels. Again both the panels are divided in 3 columns and so both the panels have 3 cells.

<table>
<thead>
<tr>
<th>ROW 1</th>
<th>ROW 1</th>
<th>ROW 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL 1</td>
<td>COL 2</td>
<td>COL 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ROW 1</th>
<th>ROW 1</th>
<th>ROW 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>COL 1</td>
<td>COL 2</td>
<td>COL 3</td>
</tr>
</tbody>
</table>

In this presentation, we will generate 2 panels and 6 cells in it and will see how different outputs will be placed in these 6 cells. To generate the cell, you need to use three ODS tagsets, one for panel, one for row and one for column. As with other ODS features, tagsets also have start and finish statements.

To create a panel, you need to start panel first, then a row and then a column as follow.

```sas
GOPTIONS Device = ActiveX;
GOPTIONS xpixels=360 ypixels=360;
ODS tagsets.HTMLpanel Path="C:\Documents and Settings\NESUG09" File = "Booksales.html";
/* Start the Panel 1 */
ODS tagsets.HTMLpanel event = panel (start);
/* Start the Row 1 */
ODS tagsets.HTMLpanel event = row_panel (start);
/* Start the Column 1 */
ODS tagsets.HTMLpanel event = column_panel (start);
```

Above mentioned code will create Booksales.html file at the location mentioned in Path option using ActiveX device. Pixel size under GOPTIONS will control the size of each cell in the panel which can be changed according to desired cell size. ODS tagsets statements with event option will start panel first, then row in the panel and then first column in the panel. You can also define your own custom style using PROC TEMPLATE and then specify that style in the STYLE option along with Path and File options in the very first ODS tagsets statement.

Now you can add any SAS code which will generate some output and it will be placed in the first cell of the panel. I am just creating a simple listing of the sales data for north region by following code.
/* Listing of Data for North region */
Proc Print Data = Booksale noobs;
   Var Region Month Quarter Sales;
   Where Region = ‘North’;
Run;

Output from the above code will be placed in the first cell (Row 1, Col 1) of the first panel. Now to create and place another output in the next cell, you must close first cell and start the next cell as follow.

/* Close the Column 1 */
ODS tagsets.HTMLpanel event = column_panel (finish);
/* Start the Column 2 */
ODS tagsets.HTMLpanel event = column_panel (start);

Above code will close the first cell and will start second cell by starting column 2. Now I will create the plot of Sales by Month for North region which will be placed in second cell (Row 1, Col 2) of the first panel as follow.

/* Plot of Sales by Month for North Region */
Proc Gplot Data = Booksale;
   Plot Sales * Month/ haxis = axis1;
   Symbol v = dot Interpol=join;
   Axis1 order = 1 to 12 by 1;
   Where Region = ‘North’;
Run; Quit;

Now you must close second cell to start third cell as follow.

/* Close the Column 2 */
ODS tagsets.HTMLpanel event = column_panel (finish);
/* Start the Column 3 */
ODS tagsets.HTMLpanel event = column_panel (start);

Now I will calculate the sales for each quarter for north region and then plot the graph of Sales by Quarter which will be placed in third cell as follow.
/* Sum of Sales by Quarter for North Region */
Proc Means Data = Booksale noprint;
   Class Quarter;
   Var Sales;
   Where Region = ‘North’;
   Output out = Stat sum = Sales;
Run;

/* Plot of Sales by Quarter for North Region */
Proc Gplot Data = Stat;
   Plot Sales * Quarter/ haxis = axis1 vaxis = axis2;
   Symbol1 v = dot Interpol=join;
   Axis1 order = 1 to 4 by 1;
   Axis2 order = 20000 to 50000 by 5000;
Run; Quit;

Output generated by above code will be placed in third cell (Row 1, Col 3) of the first panel. Now that all three cells of the first panel have been filled, you can close this panel and proceed to the next panel and create different outputs in the similar way to place in different cells of the second panel. To close the first panel, you need to follow the same steps as we did for starting the panel, but it has to be in the reverse order. First close the row, then last column of the row and then close the panel as follow.

/* Close the Row 1 */
ODS tagsets.HTMLpanel event = row_panel (finish);

/* Close the Column 3 */
ODS tagsets.HTMLpanel event = column_panel (finish);

/* Close the Panel 1 */
ODS tagsets.HTMLpanel event = panel (finish);

Above code will close the first panel. After this you can start second panel and process each cell of second panel in the similar way as we did for first panel. Following code will start second panel, will create 3 cells in it and will place different outputs in these 3 cells.

/* Start the Panel 2 */
ODS tagsets.HTMLpanel event = panel (start);
/* Start the Row 1 */
ODS tagsets.HTMLpanel event = row_panel (start);
/* Start the Column 1 */
ODS tagsets.HTMLpanel event = column_panel (start);
/* Listing of Data for North region */
Proc Print Data = Booksale noobs;
   Var Region Month Quarter Sales;
   Where Region = 'North';
Run;

/* Close the Column 1 */
ODS tagsets.HTMLPanel event = column_panel (finish);

/* Start the Column 2 */
ODS tagsets.HTMLPanel event = column_panel (start);

/* Plot of Sales by Month for South Region */
Proc Gplot Data = Booksale;
   Plot Sales * Month/ haxis = axis1;
   Symbol1 v = dot Interpol=join;
   Axis1 order = 1 to 12 by 1;
   Where Region = 'South';
Run; Quit

/* Close the Column 2 */
ODS tagsets.HTMLPanel event = column_panel (finish);

/* Start the Column 3 */
ODS tagsets.HTMLPanel event = column_panel (start);

/* Sum of Sales by Quarter for South Region */
Proc Means Data = Booksale noprint;
   Class Quarter;
   Var Sales;
   Where Region = 'South';
   Output out = Stat1 sum = Sales;
Run;

/* Plot of Sales by Quarter for South Region */
Proc Gplot Data = Stat1;
   Plot Sales * Quarter/ haxis = axis1 vaxis = axis2;
   Symbol1 v = dot Interpol=join;
   Axis1 order = 1 to 4 by 1;
   Axis2 order = 20000 to 50000 by 5000;
Run; Quit;

/* Close the Row 1 */
ODS tagsets.HTMLPanel event = row_panel (finish);

/* Close the Column 3 */
ODS tagsets.HTMLPanel event = column_panel (finish);

/* Close the Panel 2 */
ODS tagsets.HTMLPanel event = panel (finish);

In this presentation, we created a combination of tabular and graphical outputs on the same HTML page, but you can create multiple tabular outputs or multiple graphical outputs alone and
that too of different sizes on a single HTML page. You can also create a single HTML panel which has cells of different sizes in it based on your need of reporting.

The output of the two panels from above mentioned codes looks like following figure.

Also, another beauty here is the ActiveX capability. As mentioned earlier, these graphical outputs are generated using ActiveX device which gives another powerful tool to you right in your hand without any extra efforts. Once a graphical output is generated using SAS code, ActiveX gives wide range of choices for the modification of graphical output without changing a single line of SAS code. You will be amazed to see how easily you can modify a graphical output with just a click of mouse as described in following figures.
Above plot of Sales by Month for north region can be changed into a Pie chart or vertical bar chart as follow by selecting Chart Type option available from right click menu.

You can change many other countless attributes of graphical output with this capability.

Thus, a comparison of long list of graphical and tabular outputs, which might take hours of efforts easily, is greatly simplified when HTML panels are used. This method enables you to present enormous amount of information in different formats on the same page which makes comparison of even a large size data an easy task.
ODS LAYOUT

One another fine feature of SAS V9.1 is ODS LAYOUT. This feature allows you to generate multiple graphical and tabular outputs on a single page in different regions.

One limitation is that this method works best with PDF format only. ODS LAYOUT has two different ways: Absolute Layout and Gridded (relative) Layout. Gridded layout should work fine with HTML format also, but absolute layout will not support HTML format at all. Both of these methods divide the layout in regions and then put different outputs in these regions.

Absolute Layout

In absolute layout, you need to specify the horizontal and vertical space available on a page in the ODS LAYOUT statement with the use of WIDTH and HEIGHT options. Then you need to specify where to start the region in the form of x and y coordinates in the ODS REGION statement. You also need to specify the size of region in the same statement with the use of WIDTH and HEIGHT options. Size of layout and region can be specified in inches, centimeters or in the form of percentages. Same way position of region can also be specified in inches, centimeters or percentages of total space available on horizontal and vertical axes. All these measurements calculate the distance in specified unit from top left corner of the page. As any other ODS feature, layout also has start and end statements.

For presentation purpose, we will use the same data that we have used in previous section. Following code will generate a tabular output of sales, plot of sales by month and plot of sales by quarter for both north and south regions on the same page.

```
ODS PDF File = "C:\Documents and Settings\NESUG09\Booksales.pdf"
   Startpage = no;

   /* Start the Layout */
   ODS Layout start width = 9in height = 9in;

   /* Start first region */
   ODS Region width = 31pct height = 50pct x = 0pct y = 0pct;

   /* Tabular output of Sales Data for North Region */
   Proc Print Data = Booksale noobs;
      Var Region Month Quarter Sales;
      Where Region = 'North';
   Run;

   /* Start second region */
   ODS Region width = 31pct height = 28pct x = 33pct y = -80pct;

   /* Plot of Sales by Month for North Region */
   Proc Gplot Data = Booksale;
      Plot Sales * Month/ haxis = axis1;
      Symbol1 v = dot Interpol=join;
      Axis1 order = 1 to 12 by 1;
      Where Region = 'North';
   Run; Quit;
```
As specified in the above code, each region statement has size and location of region specified in the percentages of available space on the layout. All specified measurements in above code are carefully calculated manually to give enough space to each output. The drawback of absolute layout is that one output can be overlapped by another output if region specifications are not specified accurately. Also if output is big and specified region is not big enough, it will simply truncate the output without any warning.
The output of above mentioned code will look like following figure.

<table>
<thead>
<tr>
<th>Region</th>
<th>Month</th>
<th>Quarter</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>1</td>
<td>1</td>
<td>15000</td>
</tr>
<tr>
<td>North</td>
<td>2</td>
<td>1</td>
<td>15640</td>
</tr>
<tr>
<td>North</td>
<td>3</td>
<td>1</td>
<td>13250</td>
</tr>
<tr>
<td>North</td>
<td>4</td>
<td>2</td>
<td>13482</td>
</tr>
<tr>
<td>North</td>
<td>5</td>
<td>2</td>
<td>8565</td>
</tr>
<tr>
<td>North</td>
<td>6</td>
<td>2</td>
<td>9235</td>
</tr>
<tr>
<td>North</td>
<td>7</td>
<td>3</td>
<td>10532</td>
</tr>
<tr>
<td>North</td>
<td>8</td>
<td>3</td>
<td>10222</td>
</tr>
<tr>
<td>North</td>
<td>9</td>
<td>3</td>
<td>11452</td>
</tr>
<tr>
<td>North</td>
<td>10</td>
<td>4</td>
<td>12456</td>
</tr>
<tr>
<td>North</td>
<td>11</td>
<td>4</td>
<td>14694</td>
</tr>
<tr>
<td>North</td>
<td>12</td>
<td>4</td>
<td>13826</td>
</tr>
</tbody>
</table>

Gridded Layout

With gridded layout method, drawbacks of absolute method can be overcome. In this method you simply tell ODS how many columns and rows you want and it will make each grid big enough to accommodate the whole output. In this method, if you don't specify region, it will put output of each procedure in next available region from left to right and top to bottom direction. If you want specific output in a specific region then you can simply specify the region by COLUMN and ROW option when you start the region. Also you can control the size of each grid by COLUMN_WIDTHS and ROW_HEIGHTS options in ODS LAYOUT statement itself.
Output similar to above figure can be generated with gridded layout by just telling ODS that you want to have 2 rows and 3 columns in the output. This can be specified by following code.

```
ODS PDF File = 'C:\Documents and Settings\NESUG09\Booksales.pdf"
Startpage = never columns = 3;
/* Start the Layout */
ODS Layout start rows = 2 columns = 3;
```

After this code you just have to write a simple ODS Region statement before each output generating procedure without any other specifications in it and each output will be put in to next available region. The hassle of calculating the size of each region manually so that region is big enough to include the whole output is eliminated here. Also calculating and specifying the location of each region on a layout manually is not required. Though, there are other options available which still allow you to control the size and location of each grid and so you can customize your whole output.

The output from gridded layout will look like following figure.
There are many other options available in ODS LAYOUT and ODS REGION statements which can take care of many other aspects of the multiple outputs on a single PDF page. All different options can be used either in absolute or in gridded layout and detail discussion on the use of all these options is out of the scope of this paper.

CONCLUSION

As said before, SAS always offers multiple solutions to any problem and there are multiple ways to get multiple outputs on a single page. ODS has been a great help to programmers of all levels who want to enhance and customize their reports generated in day to day activity. This compilation is just an effort to put together different ways and basic concepts to generate multiple reports on a single page in your choice of destination. Some methods explained in this paper are in the experimental phase in SAS V9.1 and they have to pass through more testing. Some features might be subject to failure or unexpected result if stressed out, but you can use them as building blocks and play around at least to start with.

ACKNOWLEDGMENT

I would like to express my gratitude to Parag Shiralkar of Eliassen Group, who motivated me writing such papers and his critics and comments inspired me to improve the quality of this paper. I would also like to thank my wife Kruti, without whose patience, this paper would have not been completed. I would also like to acknowledge eClinical Solutions, division at Eliassen Group for providing the opportunity to work on this paper.

REFERENCES

- SAS V9.1 Online documentation

CONTACT INFORMATION

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

Name: NIRAJ J PANDYA
Phone: 201-936-5826
E-mail: npandya@egistar.com or niraj_mech@yahoo.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.