

An ‘Excel’lent Journey: Exploring the New ODS EXCEL Statement

Gina Huff, Western Kentucky University, Bowling Green, KY

ABSTRACT

We asked for it and once again, SAS® delivers! Now production in SAS® 9.4 M3, ODS EXCEL provides users an extremely easy way to export output to Microsoft Excel. Using existing ODS techniques that you may already be familiar with (like predefined styles, traffic-lighting, and custom formatting) in tandem with common EXCEL features (like formulas, frozen headers and footers, and page setup options) you can create a publication-ready document in a snap. As a matter of fact, the new ODS EXCEL Statement is such a handy tool you may find yourself using it for more than large-scope production programs. ODS EXCEL is so straightforward that it can be handy for quick adhoc requests as well. Join me on my adventure as I explore the features of ODS EXCEL to create institutional research reports. I offer tips and techniques you can use to make your output ‘excel’lent, too.

INTRODUCTION

The Office of Institutional Research (IR) at Western Kentucky University is responsible for providing decision-support data to key university administration including the President, Provost, deans, and department heads. IR provides a wide array of information, ranging from small data requests to large-scale research projects. Often, we are asked to provide the information in Excel format, particularly by our deans and department heads. Heretofore, it has been a rather involved process to transfer information from SAS to Excel.

Quite frankly, using tagsets or some other method for simple data requests just didn’t seem worth the time it took to complete. In these instances, I found myself using the old “copy/paste” method and then formatting the table directly in Excel before sending the information to the requestor. However, I quickly discovered that this method was much less than ideal. It seemed like nine times out of ten, the requestor would call back and want a few tweaks—“Can we add another term?” or “Can we get this table broken down by department?” There I was, back to the drawing board, having to do more and more manual formatting in Excel! That was a journey I had no desire to repeat!

For larger projects, specifically those programs that would be used on an ongoing basis, I would tackle some tagset code to get the formatting I needed. Again, however, I found this solution a little taxing. Having established a comfort level in customizing reports with formats and styles, I already had the “lay of the land” in SAS. I had not committed Excel formats and other tagset information to memory though, so I would have to review and experiment with syntax before I could begin coding the actual project. I had to add several miles to my journey just to get ready to start, but thank goodness for some exceptional SAS papers that helped me navigate and stay on track!

This leads me to the onset of ODS EXCEL. It’s almost as if SAS recognized how troublesome the road was between SAS Street and Excel Boulevard—congested and difficult with too many twists, turns, and stops along the way—so they donned their safety vests and grabbed their tools to build a new bypass that will take you straight from SAS to Excel sans obstacles. Join me on a brief journey down this new road and you may find yourself making ODS EXCEL your route of choice—it is an ‘Excel’lent new path after all!

GETTING STARTED WITH ODS EXCEL: GET ON AND START DRIVING

Perhaps the best thing about the new ODS EXCEL Statement is how straightforward it is. For those who have tried exporting to Excel with other methods, you will be particularly appreciative of the following benefits:

1. ODS EXCEL actually creates the file you want! You don’t have to transfer the XML or HTML file that SAS created to Excel format any longer. ODS EXCEL “automagically” saves the report as an Excel (XLSX) file. *No more wrong turns!*
2. You can easily create multiple worksheets within Excel. A simple option defines how worksheets should be divided. This step was not so easy with other export methods. *No more detours!*
3. The hard work you do in SAS actually shows up in the final Excel report—formats, styles, traffic-lighting, etc. You can now take advantage of the SAS syntax you have already learned for HTML, PDF, and RTF reporting to create an exceptional Excel document as well. *One road—many destinations!*

Let us begin by taking a look at how easy it is to get started. First, the Output Delivery System (ODS) uses a simple “wrapper method” to delineate what output should be saved to the file. ODS EXCEL is no different. You will initiate Excel at the beginning of the code that will create your output and then close it at the end.

```
ods excel file='c:\filename.xlsx'; Opens ODS Excel
    ...insert code here...
ods excel close Closes ODS Excel
```

Easy, right? Now let’s look at the output of a table using PROC TABULATE. All I am doing is getting a very simple cross-tab of students by level and full/part time status. You can see that with no extra effort, I have created a pretty nice looking table directly in Excel. ODS EXCEL handles column width automatically and creates output using the default EXCEL style. Additionally, it recognizes the format I specified for the values without having to use the TAGATTR option.

SAS code:

```
ods excel file='c:\SimpleOutput.xlsx';

proc tabulate data=sassets.students format=comma7.; Automatic mapping of SAS format to its Excel equivalent.
class level time;
table level="" all Blank out needless category labels to keep the table svelte and avoid redundancy.
    , time="" all;
keylabel n=" "; This removes the unnecessary “N” header. I use this every time I just need basic counts.
run;

ods excel close;
```

Excel Output:

	A	B	C	D
1		Full Time	Part Time	All
2	Graduate	1,028	1,725	2,753
3	Undergraduate	13,152	4,163	17,315
4	All	14,180	5,888	20,068

Although the default output looks quite nice, you can easily experiment with SAS styles and select your favorite. Just like the other output destinations, SAS styles provide a unique look and feel to the output—with specifications on colors, fonts, and other style considerations. And just like the other output destinations, SAS styles now work well in ODS EXCEL. I have provided a couple of my preferred SAS-supplied styles, but if you want to explore a little, run the LIST statement in the TEMPLATE procedure to get a comprehensive list of styles.

```
proc template;
list styles; Provides a list of all the styles available.
run;
```

Here is an example of assigning a style to your Excel output in ODS EXCEL.

```
ods excel file='c:\SimpleOutput.xlsx' style=excelblue;
```

Choose a different template by adding the STYLE option to the ODS EXCEL statement

style=Analysis

	A	B	C	D
1		Full Time	Part Time	All
2	Graduate	1,028	1,725	2,753
3	Undergraduate	13,152	4,163	17,315
4	All	14,180	5,888	20,068

style=Journal1A

	A	B	C	D
1		Full Time	Part Time	All
2	Graduate	1,028	1,725	2,753
3	Undergraduate	13,152	4,163	17,315
4	All	14,180	5,888	20,068

Remember that you can create a custom style as well. Use one of the SAS supplied styles as your starting point and then you can customize from there. For more information on creating your own custom templates, be sure to read papers by Lauren Haworth and Cynthia Zender (listed in the reference section). I keep both of these papers in close proximity as I refer to them often.

ROAD SIGNS: CREATING TITLES AND FOOTNOTES

Just as every good thoroughfare has ample signage, so should your reports. Therefore, SAS makes its signage—titling and footnoting—easy to customize in ODS Excel. First, let's look at adding a title using nothing more than a basic TITLE or FOOTNOTE statement.

```
ods excel file='c:\SimpleOutput.xlsx' style=pearl;
```

```
title 'Students by Level';
footnote 'Created &sysdate.';
```

Titles or footnotes can be added with the TITLE or FOOTNOTE statement.

```
...proc tabulate code...
```

```
ods excel close;
```

The default behavior is for TITLE and FOOTNOTE to display in Page Setup, not the worksheet itself.

Keep in mind that when using the above syntax, titles and footnotes will NOT appear on the worksheet while in its normal view. By default, the titles and footnotes are actually applied to the Header and Footer section and can be viewed in the Page Setup window or by selecting the Page Layout view on the View ribbon.

I prefer to have my headers display directly on the page and SAS has made that easy, too. There are several options that can be added to the ODS EXCEL statement that will help you customize your report to make it appear exactly how you want. EMBEDDED_TITLES is one such option.

```
ods excel file='c:\SimpleOutput.xlsx'
style=pearl
options ( embedded_titles='on'
embedded_footnotes='on' );
```

These simple options allow the titles and footnotes to display directly on the worksheet. The default is 'OFF'.

```
title 'Students by Level';
footnote height=8pt italic 'Created &sysdate.';
```

I added some simple style options to the footnote so that it will display in a smaller, italic font.

```
...proc tabulate code...
```

Please note that style options in the title and footnote statement will only work when the titles and footnotes are embedded. You will still have to use Excel commands to format within the Header and Footer Section.

```
ods excel close;
```

I also use the SYSDATE macro var to display current date.

	A	B	C	D
1	Students by Level			
2				
3		Full	Part	All
4	Graduate	1,028	1,725	2,753
5	Undergraduate	13,152	4,163	17,315
6	All	14,180	5,888	20,068
7				
8	<i>Created 14MAR16</i>			
9				



TIPS & TIDBITS:
Use options in the TITLE and FOOTNOTE statements to stylize your output.

You can quickly customize each TITLE or FOOTNOTE line using options.

Example of TITLE Statement:

```
title bold justify=left "Title Goes Here";
```

A few options I commonly use:

bold	
italic	
color=[any color]	text color (foreground)
bcolor=[any color]	background color
font=[select font]	font type (ex. Arial)
height=[Xpt]	font size (ex. 8pt)
justify=[location]	left, right, or center

The titles display directly on the worksheet as I prefer. Additionally, I can format the text using the style options I am already familiar with in the TITLE and FOOTNOTE statements.

By using the embedded title option, you can now use the PRINT_HEADER and/or the PRINT_FOOTER as a separate option to create the report heading/footer if you so choose. Additionally, I will demonstrate the EMBED_TITLES_ONCE option. This ensures that the title or footer only prints once rather than between each table.

```
ods excel file='c:\TwoTables.xlsx'
  style=pearl

options ( sheet_interval='none'

  print_header='WKU Office of Institutional Research'
  print_footer='Page &P of &N'

  embedded_titles='on'
  embedded_footnotes='on'

  embed_footnotes_once='on' );
```

We will discuss SHEET_INTERVAL in depth later, but in this example, I specified 'none' to ensure both tables printed on one worksheet.

You can specify the Header and Footer directly using PRINT_HEADER/FOOTER and reserve your TITLE and FOOTNOTE statements for titling within the report itself with the EMBEDDED_TITLES/FOOTNOTES options.

Use the EMBED_FOOTNOTES_ONCE option to limit the footnote to the end of the worksheet. There is also an EMBED_TITLES_ONCE option.

```
title bold height=10pt color=firebrick 'Level and Location';
footnote height=8pt italic 'Created &sysdate.';
```

```
proc tabulate data=sassets.students format=comma7.;
  class level time onoffcampus;
  table level='', time='' all, onoffcampus='' all;
  keylabel n = ' ';
run;

ods excel close;
```

I have modified the TABULATE code so that it now produces two tables (one for graduates and one for undergraduates).

	A	B	C	D
1	Level and Location			
2				
3	Graduate			
4		Off Campus	On Campus	All
5	Full Time	1,007	21	1,028
6	Part Time	1,718	7	1,725
7	All	2,725	28	2,753
8				
9	Undergraduate			
10		Off Campus	On Campus	All
11	Full Time	8,179	4,973	13,152
12	Part Time	4,129	34	4,163
13	All	12,308	5,007	17,315
14				
15	<i>Created 14MAR16</i>			

The PRINT_HEADER and PRINT_FOOTER option display at the very top and bottom of your worksheet.

Level and Location			
Graduate			
	Off Campus	On Campus	All
Full Time	1,007	21	1,028
Part Time	1,718	7	1,725
All	2,725	28	2,753
<i>Created 14MAR16</i>			
Undergraduate			
	Off Campus	On Campus	All
Full Time	8,179	4,973	13,152
Part Time	4,129	34	4,163
All	12,308	5,007	17,315
<i>Created 14MAR16</i>			

The footnote would print under each table if we didn't use the EMBED_FOOTNOTES_ONCE option.



TIPS & TIDBITS:
Specialty formatting within the Header/Footer of Excel

You cannot use SAS style options to change formatting within the Header or Footer, but you can use Excel controls to do so. There is one drawback, however. Because the Excel controls also begin with a leading ampersand (&), you do receive a warning in the log. You also want to make sure that you don't create a user-defined macro var with the same name because the SAS macro variable will trump the Excel control.

Example of ODS EXCEL Option:

```
options (print_footer='Page &P of &N')
```

Options I commonly use:

Justification	
&C	Center
&R	Right
&L	Left
Pagination	
&P	Page Number
&N	Number of Pages
Identification	
&A	Sheet Name
&D	Date
&T	Time

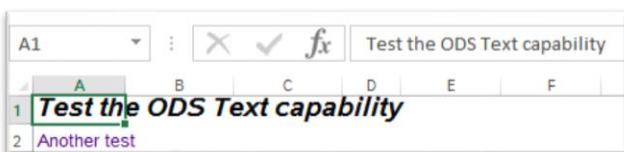
ODS TEXT is another very flexible way of inserting text into your document. Consider using this statement to add incidental information into your worksheet. Have even more control over the appearance of the output by using inline formatting options (ODS ESCAPECHAR). You must first use the ODS ESCAPECHAR statement to identify the character that initiates the inline format (I used the caret [^] in the example below). Once that is initiated, you can use a myriad of style options to customize the text. (See Huff, 2011 and 2014 for further style tips.)

Keep in mind, however, that ODS TEXT merely places the new text in the next available cell. It does not merge across the cells to accommodate the report as the TITLE statements do. This may be a consideration as you design your report.

ODS TEXT also only prints after the last procedure. If, for example, you decide to output to tabs by “BYGROUP”, then the ODS TEXT will not print on every page. Instead, it will print on the last page—after the procedure is complete. Titles and footnotes, on the other hand, will print on every page unless the EMBED_FOOTNOTES_ONCE option is activated. This may play in to your decision when selecting how to display text.

```
ods excel file='c:\TwoTables.xlsx'
      style=pearl;
ods escapechar='^';      Initialize the escape character to use for inline formatting.
ods text='^{style[font_size=14pt font_weight=bold font_style=italic]
      Test the ODS Text capability}';
ods text='^{style[font_size=10pt foreground=indigo]Another test}';
ods excel close;
```

Specify desired style formats.



When using ODS TEXT, the text is created and placed in the next available cell. Cells are not merged across the report.

EASY ACCESS TO FAMILIAR SAS DESTINATIONS

ODS Excel affords you the opportunity to put your knowledge of SAS Syntax to instant use. The ODS EXCEL statement translates SAS options, formats, and styles with ease.

SAS Options:

You can assign page layout options within the system OPTIONS statement and it will seamlessly convert to Excel with one exception: you CANNOT assign orientation in the SAS OPTIONS statement. Don't ask me why. I just know that I made a wrong turn when trying to assign it there and ended up having to make a U-turn and put orientation back into the ODS EXCEL options. Hopefully, this little tip will help you avoid getting lost as well.

```
options leftmargin=0.5in
        rightmargin=0.5in
        topmargin=1.0in
        bottommargin=1.0in
        papersize=legal;
```

Define page margins

Specify paper size (legal=8½ x 14).

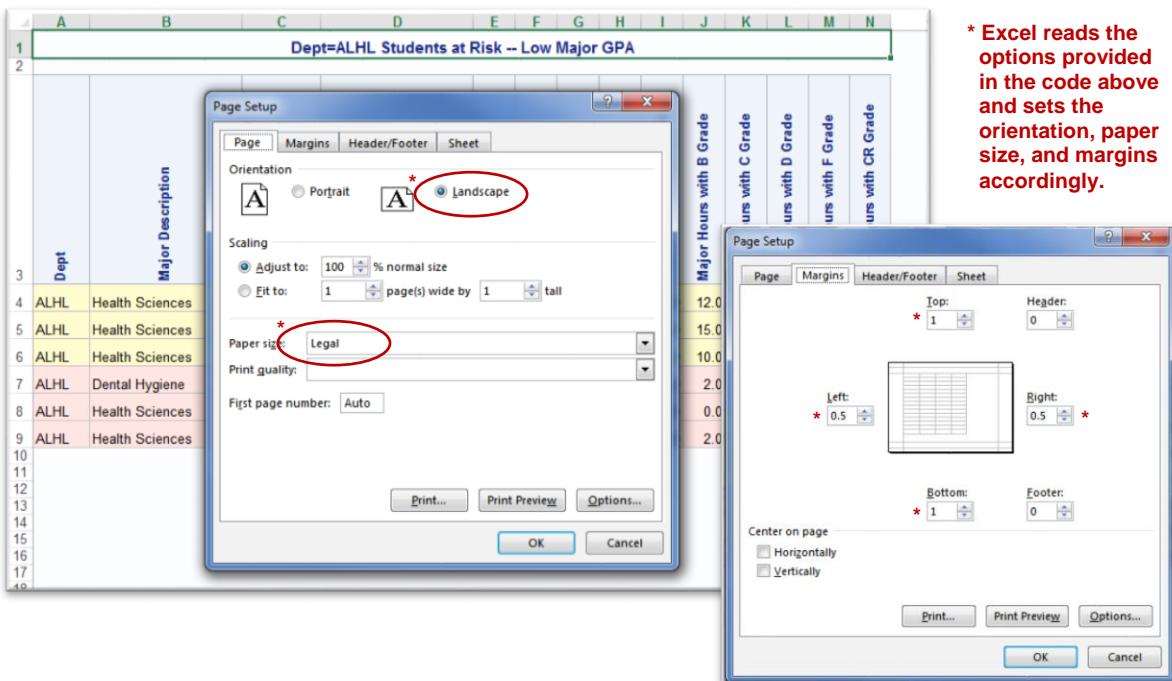
```
ods excel file='c:\risk report.xlsx'
  options( embedded_titles='yes'
          embedded_footnotes='yes'
          orientation='landscape'
          ...other options... );
```

Note: The 'YES' alias also works to initiate embedded titles/footnotes.

Specify page orientation in the EXCEL options, NOT the SAS system options. Portrait is the default setting.

...title statement and report procedure...

```
ods excel close;
```



* Excel reads the options provided in the code above and sets the orientation, paper size, and margins accordingly.

Formats:

SAS has already done much of the heavy lifting in assigning formats so you don't have to know Excel formats any longer. Instead, ODS EXCEL automatically maps the SAS format to its Excel equivalent. However, as with any new system, there are a few quirks that we will touch on and troubleshoot. Here are a few examples and we will quickly drive on...

First, I created a table that had nothing more than three variables with the same SAS date (20826) to demonstrate how easy it is to display dates in various formats. BTW, the date just happens to be my upcoming birthday, so mark your calendars and send me a quick email on January 7 if you found this paper of value to you throughout the year.

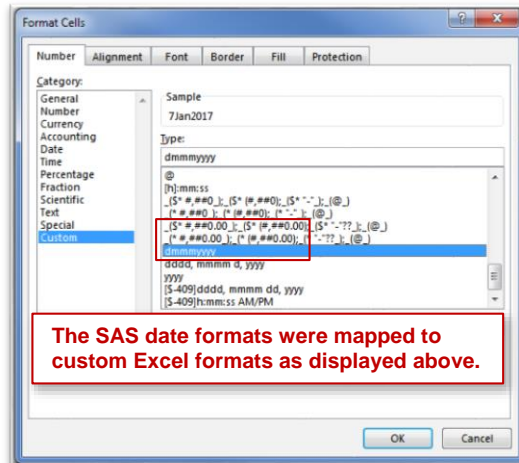


TIPS & TIDBITS: Find a format for all of your needs

SAS documentation on every for every SAS 9.4 format can be found at <http://support.sas.com/documentation/cdl/en/leforinforref/64790/PDF/default/leforinforref.pdf>



```
ods excel file='c:\format examples.xlsx' style=festival;
data numbers;
    date1='07jan2017'd;
    date2='07jan2017'd;
    date3='07jan2017'd;
run;
proc print noobs;
    var date1 date2 date3;
    format    date1 date9.
              date2 weekdate29.
              date3 year4. ;
run;
ods excel close;
```



	A	B	C
1	date1	date2	date3
2	7Jan2017	Saturday, January 7, 2017	2017

Next, I have provided a few examples of currency. Let's look at the format for the number 2500 as our example. We know that when using SAS formats, if we set the format to dollar4. the output would read "2500". That's because SAS takes out all non-essential items (like commas and dollar signs) to keep the value at the assigned format length of four. When you send that to Excel, however, SAS maps the format as currency, so both the dollar sign (\$) and comma (,) show up in the Excel output. If you set the format to dollar4.2, the format would still be mapped as currency so the dollar sign and comma would be present but because the size is too small, the decimal places would not. A minor thing, but important when understanding how SAS formats map to Excel. Best practice is, of course, to set the format at the appropriate length so that you don't have to worry about these little issues.

```
ods excel file='c:\format examples2.xlsx' style=money;
data numbers;
    price1=2500;
    price2=2500;
    price3 =2500;
run;
proc print noobs;
    var price1 price2 price3;
    format    price1 dollar4.
              price2 dollar4.2
              price3 dollar9.2;
run;
ods excel close;
```

	A	B	C
1	price1	price2	price3
2	\$2,500	\$2,500	\$2,500.00

price1	price2	price3
2500	2500	\$2,500.00

Notice how dollar formats map as currency in Excel (above), but decimal places only show up if you have designated an adequate length. At left is the output as it would appear in html.

The final formatting example is little trickier. Percent formats seem to have a slight glitch in the new system. When I try to output percents with decimal places, even with ample space, the mapped Excel formats still only give me the percent with no decimal allowance. I can work around this by using a picture format to achieve the desired output, but that solution is less than ideal. Let's hope the next release solves this issue.

```
proc format;
    picture mypercent low-high='99.99%' (mult=10000);
run;
```

I use a picture format to solve the decimal place dilemma. It's not the ideal solution, but it works. You will see that I reference this format in the last example (percent4).

```
ods excel file='c:\format examples2.xlsx' style=statistical;
data numbers;
  percent1=.9876;
  percent2=.9876;
  percent3=.9876;
  percent4=.9876;
run;
proc print noobs;
  var percent1 percent2 percent3 percent4;
  format   percent1 percent6.
          percent2 percent6.2
          percent3 percent20.2
          percent4 mypercent.;
run;
ods excel close;
```

	A	B	C	D
1	percent1	percent2	percent3	percent4
2	99%	99%	99%	98.76%

Excel output did not output decimal places with the percent format. The workaround was using a picture format instead.

percent1	percent2	percent3	percent4
99%	99%	98.76%	98.76%

Above, html output is displayed. The format assigned to percent3 does output decimals properly ordinarily.

Percent4 uses the picture format created above.

Traffic Lighting:

Traffic Lighting is an excellent way to draw attention to important elements within your report and it is a technique I use often when creating reports for PDF or HTML output. I can use the same syntax and get equivalent results in ODS Excel. Here is an example of traffic lighting using two different methods. In the example, I am developing a report that provides a list of at-risk students for departments to use when counseling and advising their students. If a student's GPA in his/her major course of study is less than 2.20 then that student is deemed "at risk". If the student has a GPA of less than 1.75 then he/she is considered high risk and are highlighted in light red. Additionally, if the student's overall GPA is also a concern, then the overall GPA needs to be emphasized also.

```
proc format;
  value gpa_color
    low - 2.20="firebrick"
    2.20 <- high="black";
  value gpa_weight
    low - 2.20="bold"
    2.20 <- high="medium";
run;
```

The first format will define the text color for GPA. If the GPA is 2.20 or lower, then firebrick (a deep red) is defined. Otherwise, the font color will be basic black.

The second format designates font weight in the same fashion.

```
proc report data=redflag nowd
  style(header)={tagattr='rotate:90'}
  style(column)={cellwidth=500};
by dept;
column dept desc id name hrs_earned hrs_attempted gpa maj_gpa
  hrs_a hrs_b hrs_c hrs_d hrs_f hrs_cr ;
  define gpa / style={foreground=gpa_color.
                    font_weight=gpa_weight.};

compute maj_gpa;
  if maj_gpa.sum < 1.75
    then call define (_row_, "style", "style={background=mistyrose}");
  else if maj_gpa.sum < 2.20
    then call define (_row_, "style", "style={background=lemonchiffon}");
  else call define (_row_, "style", "style={background=white}");
endcomp;
run;
```

I use the formats I created above in the style option to determine the color of the text (foreground) and the weight of the font (font_weight)

I use the compute statement to define the row background color based on the value of MAJ_GPA.

Here's an example of the resulting output:

Dept=CFS Students at Risk -- Low Major GPA													
Dept	Major Description	ID	Name	Hours Earned	Hours Attempted	GPA	Major GPA	Major Hours with A Grade	Major Hours with B Grade	Major Hours with C Grade	Major Hours with D Grade	Major Hours with F Grade	Major Hours with CR Grade
CFS	Interior Design and Fashion Merchandising	800xxxxxx	Txxxxxxxx,Jxxxxxx	95	9	2.64	1.00	0.0	3.0	0.0	0.0	6.0	0.0
CFS	Hospitality Management and Dietetics	800xxxxxx	Sxxxxxxxx,Exxxxxx	142	46	2.20	2.07	3.0	12.0	22.0	3.0	6.0	0.0
CFS	Hospitality Management and Dietetics	800xxxxxx	Bxxxxxxxx,Jxxxxxx	141	51	2.64	1.76	0.0	8.0	26.0	14.0	3.0	0.0
CFS	Family and Consumer Science Education	800xxxxxx	Jxxxxxxxx,Dxxxxxx	100	54	2.42	2.11	6.0	12.0	27.0	0.0	9.0	0.0
CFS	Hospitality Management and Dietetics	800xxxxxx	Jxxxxxxxx,Nxxxxxx	94	44	2.26	2.07	3.0	6.0	26.0	9.0	0.0	0.0
CFS	Interior Design and Fashion Merchandising	800xxxxxx	Gxxxxxxxx,Bxxxxxx	83	3	2.20	1.26	0.0	3.0	12.0	6.0	10.0	0.0

Proper shading based on the Major GPA

Additional emphasis on those with a low overall GPA

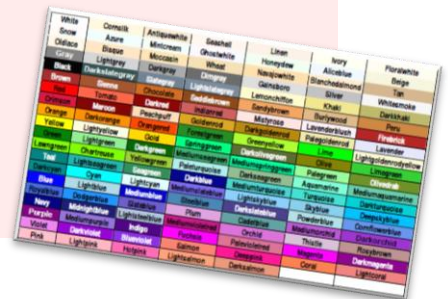
If you want to delve a little deeper into the traffic lighting process, there is a helpful paper by Louise Hadden listed in the reference section to provide a lot more detail than I can do in this brief example. Suffice to say, though, traffic lighting is a power tool to use when presenting data and emphasizing important elements. Learning these techniques will be of great value when reporting out to many of the ODS destinations, including ODS EXCEL.



TIPS & TIDBITS: Using Color in ODS Excel, the Rainbow at Your Fingertips

SAS ODS offers a full spectrum of color choices, supporting the Hex RGB (Red/Green/Blue) color palette. To get the Hex value for a color, SAS Support directs users to visit <http://the-light.com/colclick.html> where you can pick a color and immediately get the corresponding Hex value.

SAS also has a robust listing of color names that you may use. I have found the color table provided by Lauren Haworth in her SUGI 30 paper *SAS with Style: Creating your own ODS Style Templates for RTF Output* a valuable resource (<http://www2.sas.com/proceedings/sugi30/132-30.pdf>). Print it and place it where it is easily accessible. That way you will know exactly how the printed color will appear, as it is often quite different than the way the color appears on screen. I have found the colors named and provided by SAS are adequate to handle the majority of my output needs.



EASY CONNECTIONS TO OTHER STREETS: EXCEL FEATURES

Excel has many options as a spreadsheet utility that are different than a word processing (RTF) or portable document format (PDF). If you are used to programming ODS PDF or ODS RTF then you will want to familiarize yourself with some of these new streets. If you have already been programming using tagsets, however, these routes may already be familiar to you.

Page Layout:

First, let's look at some simple ways to customize the way that Excel sizes and displays the data on the page. As you probably know if you have worked in Excel, you can center your table(s) and scale the size of your table(s) to fit within a page(s). ODS EXCEL options allow you to take advantage of these page layout features.

CENTER_VERTICAL and CENTER_HORIZONTAL are two options that are the equivalent of the *Center on Page* option in the *Page Setup* menu of Excel. The default is 'off', but if you have a small table, you may want to switch the horizontal option to 'on' so that the table centers on the page. Similarly, if you want the table centered vertically on the page, then you would use the CENTER_VERTICAL option.

```
ods excel file='c:\SimpleOutput.xlsx'
      style=pearl
      options ( center_horizontal='on'
                embedded_titles='on'
                embedded_footnotes='on');

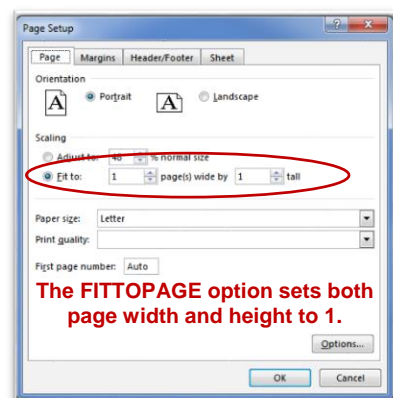
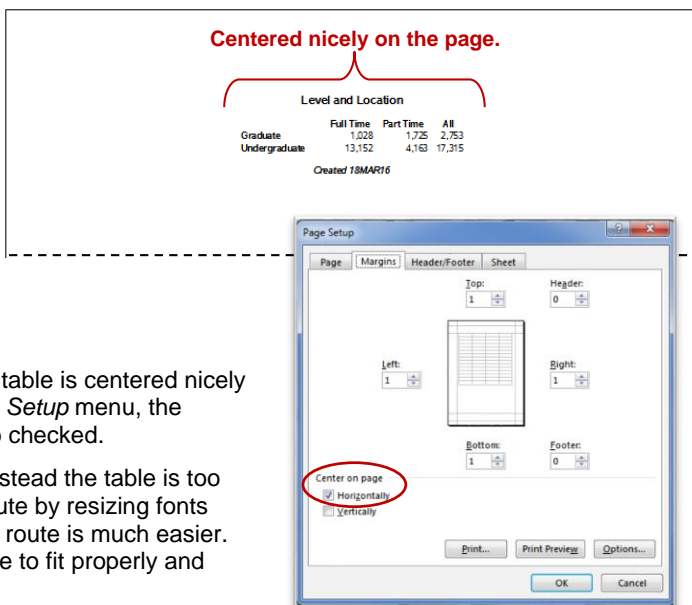
      ...title statement
      and tabulate procedure...
```

```
ods excel close;
```

You can see that now when we look at a print preview, the table is centered nicely on the page. Also, when we check the options in the *Page Setup* menu, the Center on Page Horizontally checkbox has been toggled to checked.

Oftentimes, the issue isn't with the table being small, but instead the table is too big to fit on the page. You could, of course, go the SAS route by resizing fonts and working with style options to force a fit, but the EXCEL route is much easier. With simple options in ODS EXCEL, you can scale the table to fit properly and print the number of pages you specify.

If you just want the table to fit to one page, then you can set the option FITTOPAGE to 'on'. If, however, you want the table to fit to a prescribed number of pages, then the PAGES_FITWIDTH and PAGES_FITHEIGHT are powerful tools.



FITTOPAGE example (at left):

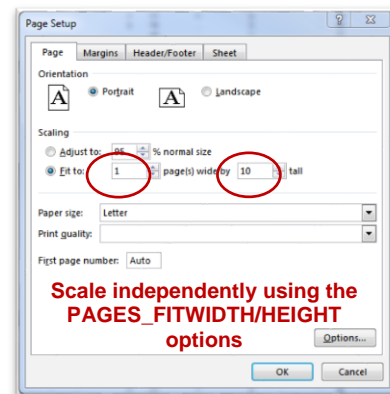
```
options ( fittopage='on' )
```

FITWIDTH/HEIGHT example (at right):

```
options ( pages_fitwidth='1'
          pages_fitheight='10' )
```

scale option (not pictured):

```
options ( scale='90' )
```



If you know the exact percentage you would like the table scaled (say you wanted it to be 90% of normal size), SAS also has an option that will handle that. To no surprise the option is called "scale" and I have provided sample syntax above. I don't often know the exact percentage I want the table resized, so I find the fitwidth/height options more helpful.

When the table outputs to more than one page, chances are that you will want the titles and column headers to print on each of the pages. Again, options in ODS EXCEL make that an easy task.

```
options ( row_repeat='header' )
```

If you have a specific number of rows that you want to repeat, then you may also select a range. For example, if you wanted the first 5 rows to repeat on each page, then the syntax would read, `row_repeat='1-5'`. As you would expect, SAS offers the COLUMN_REPEAT option that will designate the columns that may need repeating across pages.

Table Layout:

There are several ODS EXCEL preferences that help enhance table layout and functionality using the options statement. I often use row height and column width. These options allow you to specify just how wide or how high the columns and rows will be. The example provided exaggerates the widths and heights so that you can clearly see the option's behavior.

```
ods excel file='SimpleOutput.xlsx' style = htmlblue
options ( absolute_row_height='30' Because I listed only one value, every row will be assigned a height of 30.
          absolute_column_width='60, 10' You can also specify each row or column individually. In
          embedded_titles='on'           column_width, I specified TWO values. Because there are more
          embedded_footnotes='on' );     than two columns in the table, SAS will assign the widths
                                         sequentially: the first column will be assigned a width of 60, the
                                         second will be assigned a width of 10, the third will go back to 60,
                                         the fourth 10, and so on.

...title statement
and tabulate procedure...

ods excel close;
```

Level and Location			
	Full Time	Part Time	All
Graduate			
Undergraduate	1,028		1,725 2,753
	13,152		4,163 17,315

Remember that SAS will automatically assign what it deems the best width and height for columns and rows. For basic tables, this will usually suffice. However, as you begin to develop more customized reports, the need for added control is needed. Use ABSOLUTE_ROW_HEIGHT/WIDTH to better customize your table.

The ODS EXCEL statement has one other option that is quite handy as it relates to row widths. It is called ROW_HEIGHTS and it actually lets you control height based on the row type. For example, the first position will set the height for header rows, the second position will set height for body (or data) rows, etc. I find this option more helpful than ABSOLUTE_ROW_HEIGHT when I need to assign more than one height because you can assign it based on the type of row it is rather than having to assign every row individually. The table below provides the positional values.

First	Second	Third	Fourth	Fifth
Header	Body	BY Lines	Titles	Footers

```
options ( row_heights = '150, 20, 20, 20, 12' )
```

There are many other table options that you can use to enhance your output and take advantage of EXCEL formatting features. Among them are frozen headers, hidden columns and rows, and auto-filters. Additionally, you can specify the starting position for your table (doesn't always have to start in cell A1), and suppress by-lines within the options of the ODS EXCEL statement.

The following table provides a brief list of some of my go-to options. Take a moment to review the SAS User's Guide for a complete list of features.

Body rows(2) have a height of 20

Dept	Major Description	ID
CFS	Interior Design and Fashion Merchandising	800xxxxxx
CFS	Hospitality Management and Dietetics	800xxxxxx
CFS	Hospitality Management and Dietetics	800xxxxxx
CFS	Family and Consumer Science Education	800xxxxxx
CFS	Hospitality Management and Dietetics	800xxxxxx
CFS	Interior Design and Fashion Merchandising	800xxxxxx

Option Name	Choices (default value listed first)	What the Option Does
autofilter	'none', 'all', '<range>'	Turns on the autofilter option in Excel
frozen_headers	'off', 'on', '<range>'	Keeps headers stationary so they will not scroll with the scroll bar
hidden_columns	'<number, list, or range>'	Hides the columns specified
hidden_rows	'<number, list, or range>'	Hides the rows specified
start_at	'<column>, <row>' (ex. '1,1')	Specifies the starting position for output
suppress_bylines	'off', 'on'	Turns off/on the printing of bylines in output
title_footnote_width	'0', '<number of columns>'	Specifies the number of columns that titles and footnotes should span—zero indicates the number of columns used in the worksheet

Sheets/Tabs:

ODS EXCEL affords you the opportunity to put tables into different tabs within your Excel workbook. The SHEET_INTERVAL option specifies when a new worksheet is created. Additionally, there a couple of valuable options that will allow you the ability to customize the way tab labels will appear.

First, the SHEET_INTERVAL option delineates when a new worksheet will be created in the active Excel workbook. The default is to create a new worksheet after every new table, but that is rarely the way I need the output presented. Let's look at an earlier example to illustrate. The table at left reflects the output I desire—two tables on the same worksheet, but the default action is to output two tables on two separate worksheets, like the output on the right.

The first table (far left) displays the information as we would like for it to appear. We can achieve that selecting 'none' in the SHEET_INTERVAL option.

The second table illustrates the default behavior. A three-dimensional PROC TABULATE table will have a separate sheet for each page because the default behavior is to create a new tab for every new table. In this example graduate information is on Tabulate 1 and undergraduate data can be found on the second tab, Tabulate 2

Revisit the code on page 4 to see the SHEET_INTERVAL option in action.

The other choices you can make besides 'none' when deciding how to output you data include 'page', 'proc', and 'bygroup'. The default choice is 'table'.

The example above illustrated how condense output into a single worksheet, but we often want to organize output into tables. When we do, SHEET_INTERVAL will allow you to separate on page, procedure, or bygroup. Remember, too, that you can change options even within the same ODS EXCEL file, so don't be afraid to toss in another ODS EXCEL statement and change your options as needed.

When we do need reports organized into multiple tabs, very rarely are the default tab names adequate. "Tabulate1-Table1" is not very helpful. That is where SHEET_NAME can be very handy. I would recommend adding these options to your code any time you create a multi-sheet workbook in Excel.

In the example that follows, I am creating more descriptive labels for my worksheets. Unfortunately, I could not find an easy way to label the tabs based on the page dimension of PROC TABULATE output, but a change to BY groups allowed me to by-pass the road block.

```
ods excel file='c:\TwoTables.xlsx' style=pearl
  options ( sheet_interval='bygroup'
            sheet_name = '#byval(level)'
            embedded_titles='on' );
```

In this case, I want two sheets, one for each BYGROUP
The SHEET_NAME option will print the name of the applicable value – Graduate or Undergraduate.

```
title bold height=10pt color=firebrick 'Level and Location';

proc sort data=sassets.students out=students; by level;
run;

proc tabulate data=students format=comma7.;
by level;
  class time onoffcampus;
  table time=' ' all, onoffcampus=' ' all;
  keylabel n = ' ';
run;

ods excel close;
```

Data must be in the proper sort order for a BY statement to work

I modified the syntax a bit to get the tabs to label the way I wanted them to. Instead of creating a three-dimensional table, I used the BY statement to create the separate Graduate and Undergraduate tables.

In the output, you can see that the tabs are now given a much more descriptive name.

Level and Location			
Level Code=Undergraduate			
	Off Campus	On Campus	All
Full Time	8,179	4,973	13,152
Part Time	4,129	34	4,163
All	12,308	5,007	17,315

The tabs are given a descriptive name to better organize your Excel workbook.

In the next example, I add another option that will assist you in customizing tabs: the TAB_COLOR option.

As I mentioned previously, you can use ODS EXCEL Options multiple times within the same output. I also demonstrate that capability by creating three frequencies using PROC FREQ and labeling each individually.

```
ods noproctitle;
ods excel file='c:\TwoTables.xlsx' style=pearl
  options ( embedded_titles='on'
            embedded_footnotes='on'
            sheet_name = 'Level' );
```

Use the NOPROCTITLE option to suppress the SAS automatic title 'The FREQ Procedure'
The first tab will be labeled 'Level' based on the first options statement.

```
title bold height=10pt 'Students by Level';
proc freq data=sassets.students;
  table level / norow nocol;
run;
```

```
ods excel options (sheet_name='Time Status');
title bold height=10pt 'Students by Time Status';
proc freq data=sassets.students;
  table time / norow nocol;
run;
```

The next tab will be named 'Time Status'. All other Excel Options are retained.

```
ods excel options ( sheet_name='On/Off Campus'
                  tab_color = 'red');
title bold height=10pt 'Students by Campus Residency';
proc freq data=sassets.students;
  table onoffcampus / norow nocol;
run;
ods excel close;
```

In the last tab, the sheet name will change AND the color of the tab will be highlighted in red using the TAB_COLOR option.

The resulting output shows that the workbook is organized to the nines with just a few extra stops along the road.

LEVEL	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Graduate	2753	13.72	2753	13.72
Undergraduate	17315	86.28	20068	100.00

Tabs are aptly titled using SHEET_NAME and one tab is highlighted in red for quick recognition using the TAB_COLOR option.

EXPLORE NEW PATHS: EXCEL FORMULAS AND GRAPHING CAPABILITY

The key benefit of working in Excel (beyond PDF or word processing output) is in its unique ability to work with numeric data, namely in its ability to use formulas and create graphical display. It's also accessible and familiar to a host of clientele, making it a desirable output destination. In a session I attended last year, the speaker summed it up well by saying that he has to output to Excel because that's "where the business world likes to live." Our institution is no different. We get several requests for data in Excel and now with ODS EXCEL, output is easy—as evidenced above. However, we now have the capability to take things a step further and create production reports that include formulas and even native Excel graphs (not just images). For the next section, we will travel down this new road a bit.

The example I will be using for the purpose of this portion of the paper will be to report items relevant to WKU's Composite Academic Index (CAI). This is the index used by admissions to determine a first-time applicant's acceptance into the university. CAI is calculated by taking the student's high school GPA multiplied by 20 and then adding the student's ACT score.

Excel Formulas:

If we wanted a detail report listing a student's CAI, we could certainly do so using SAS. However, what if the requestor wanted to be able to modify the scores and see how that would impact the potential student's CAI score? What if the student increase his ACT score by 3 points, for example? Simple output would only produce data, not Excel formulas. We can use the power of tagsets in ODS EXCEL. In the following example I use PROC Report to create a detail list of students with their CAI scores. For the purpose of this example, I limited the output to only ten observations and redacted any identifying information.

Academic Index (CAI) contains a formula that calculates the value in Excel.

	A	B	C	D	E
1	WKU ID	name	High School GPA	ACT Composite Score	Academic Index
2	800xxxxxx	Sxxxxxxx,Jxxxxxx	1.92	28	66.4
3	800xxxxxx	Txxxxxxx,Dxxxxxx	2.03	20	60.6
4	800xxxxxx	Yxxxxxxx,Bxxxxxx	2.76	22	77.2
5	800xxxxxx	Bxxxxxxx,Cxxxxxx	2.28	31	76.6
6	800xxxxxx	Cxxxxxxx,Mxxxxxx	3.74	24	98.8
7	800xxxxxx	Jxxxxxxx,Cxxxxxx	3.87	30	107.4
8	800xxxxxx	Oxxxxxxx,Sxxxxxx	2.93	24	82.6
9	800xxxxxx	Mxxxxxxx,Bxxxxxx	2.44	18	66.8
10	800xxxxxx	Fxxxxxxx,Rxxxxxx	4.00	29	109.0
11	800xxxxxx	Gxxxxxxx,Txxxxxx	3.58	27	98.6

```
ods excel file='c:\formulas.xlsx';
options obs=10;
proc report data=students;
  column dw_id name gpa act CAI;
  define CAI / "Academic Index"
    computed
    format=5.1
    style={tagattr="formula: (RC[-2]*20+RC[-1]) "};
  compute cai;
    cai=0;
  endcomp;
run;
ods excel close;
```

CAI is not in the data set "STUDENTS". It is a column created to house the formula needed to send to Excel.

Use STYLE= and TAGATTR= to create the formula. RC[-2] indicates that the first value will be found two cells to the left and RC[-1] indicates the second value can be found one cell to the left.

Excel has to read this column in as a numeric and the value CANNOT be missing (.). Setting CAI to 0 is a necessary step to successfully create a formula that can be interpreted by Excel.

Now, the admissions counselor can easily go in and update either the ACT or GPA to determine how an increase in scores would impact placement. This is just one example to illustrate the capability of ODS EXCEL to give you some ideas on what can be done in ODS Excel. I wish I could expand further, but as this is a paper and not a book, I will have to cut our journey short. Keep in mind that there have been several SAS papers written on tagsets that include greater detail on formulas and can provide further guidance. One paper that serves as a quick guide is by Eric Gebhart and is included in the reference section (go straight to the section titled Hide & Seek, Formulas).

Native Excel Graphs:

It may be preproduction, but PROC MSCHART has already made its way to my list of commonly used procedures. It is an incredibly straightforward, easy way to create graphs in Excel. And these aren't just images that are saved in Excel. They are native Excel Graphs that can be edited and modified directly in the spreadsheet! Let's look at an example so that you can marvel at how easy it is too.

Again using student test score data, I decided to provide the average ACT of incoming students by their major college. Because I am beginning with the detail data, I have to first aggregate the data into the average ACT by college and then I can feed that into PROC MSCHART. The syntax follows.

```
ods excel file='c:\graphs.xlsx';
proc means data=students noprint;
  by college;
  var act;
  output out=act_avg mean(act)=average_act;
run;
```

Step 1: Prep the data. Aggregate if needed.

```

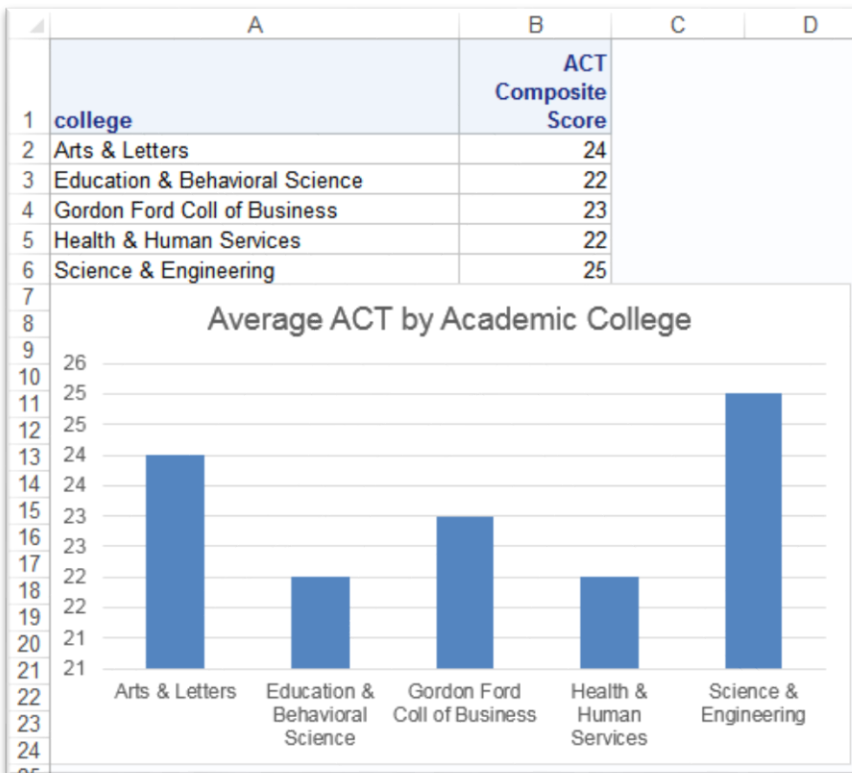
title 'Average ACT by Academic College';
proc mschart data=act_avg category=college;
  vcolumn average_act;
run;
ods excel close;

```

Step 2: Use MSCHART procedure to create your Excel Graph and Voila! Native Excel Chart!

Notice that PROC MSCHART syntax is very straightforward. You select a category (if applicable) on the PROC MSCHART statement and then you select one of several chart types including VCOLUMN (as shown above), HCOLUMN, LINE, PIE, and SCATTER. You can even create a combination chart by selecting multiple chart statements. If, for example you needed to create a chart that had both a bar and line graph, then you would use one VCOLUMN statement and one LINE statement. SAS will handle the rest.

SAS handles the rest and maps it to Excel so that it reads it as an Excel graph. The output look quite nice for such simple code and no additional style specifications or formatting.



Please note that because this is pre-production, you will get a warning message in your log stating, "MSCHART is a preproduction procedure and currently works only with the ODS destination for Excel." We are in luck, though, because ODS EXCEL is exactly where we are using the procedure and even pre-prod the output looks great.

Again, this is just a glance at the procedure and how it can enhance your Excel reports, but I hope you are energized about how useful this new procedure can be. SAS has done a pretty nice job providing documentation for the procedure even in its preproduction form. It is located in Chapter 7 of the SAS 9.4 Output Delivery System: Procedures Guide and I have provided the link in references.

Sparklines:

Sparklines are a popular way to provide information at-a-glance. You can use the method below to create quick sparklines in your Excel table as an alternate way to graph data in your Excel output. In the example below, I have created a formula as a new variable rather than a tagset attribute (tagattr=). This is another way that you can create formulas to output to Excel. Keep in mind, however, that this method does not use row/column references, so if the data were changed once it is output to Excel, then the table would not automatically change unless the formula was also updated.


```
ods excel file='c:\Sparklines.xlsx'
      options (absolute_row_height = '20');

proc freq data=freshmen noprint;
  table college / norow nocol nopercnt
      out=freshman_coll1;
run;

data freshman_coll2;
  set freshman_coll1;
  formula = '=REPT("|", '||strip(count/10)||)';
run;

proc print data=freshman_coll2 noobs label;
  var college count;
  var formula / style={foreground=firebrick font_face='playbill'};
  label    formula = "00"x
          count = "Headcount";
run;

ods excel close;
```

First I create a frequency count of our first-time freshman by the academic college of their first major.

In this step, I create an Excel formula that will repeat the designated text “|” for a prescribed number of times. This will determine the size of the bars. In this case, the number of repetitions will be the value in the variable COUNT that was created from the frequency above. I divide by 10 to decrease the overall size of the column but keep the correct scale.

I define the color of the lines using style options within my PROC PRINT. I also define the font as Playbill because the font is thick enough to make the lines overlap and look like a continuous bar. If you used another font, you can see the space between each line.

I use “00”x to make the header row for the formula column blank. If you don’t make it blank, then the header row will also be red and in Playbill font. This isn’t a readable font, so I always opt to blank out the header row using this method.

	A	B	C
1	College	Headcount	
2	Arts & Letters	538	██████████
3	Gordon Ford Coll of Business	302	██████
4	Education & Behavioral Science	215	████
5	Health & Human Services	599	██████████
6	Science & Engineering	627	██████████

I hope this brief example will “spark” your own imagination on how you can use ODS EXCEL to customize your output and best fit your needs.

CONCLUSION

There is no doubt that SAS has the power to create effective publication-ready reports using some of the flexible and powerful features we have grown to love and expect out of SAS. Now it is even easier to output reports into a popular and common destination: Microsoft Excel. By incorporating style elements that you are already familiar using in other destinations like PDF and RTF, the new ODS EXCEL utility eliminates the time-consuming and error-prone step of formatting, stylizing, and sizing cells, Separating tables into separate tabs, manually inputting formulas, and manually creating charts and graphs.

Although I just touched the surface as it relates to much of the functionality of this new statement, I hope I peaked your interest to have your own ‘Excel’-lented journey using SAS ODS EXCEL.

REFERENCES

ODS ESCAPCHAR:

Zender, Cynthia L. (2007). "Funny ^Stuff~ in My Code: Using ODS ESCAPECHAR." *Proceedings of the 2007 SAS Global Forum Conference*. Available at <http://www2.sas.com/proceedings/forum2007/099-2007.pdf>.

ODS EXCEL:

Parker, Chevell (2015). "Staying Relevant in a Competitive World: Using the SAS Output Delivery System to Enhance, Customize, and Render Reports." *Proceedings of the 2008 SAS Global Forum*. Available at <http://support.sas.com/resources/papers/proceedings15/SAS1880-2015.pdf>.

Parker, Chevell (2014). "Secrets from a SAS Technical Support Guy: Combining the Power of the SAS Output Delivery System with Microsoft Excel Worksheets." *Proceedings of the 2014 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings14/SAS177-2014.pdf>.

SAS (2015). *SAS 9.4 Output Delivery System: User's Guide*. Chapter 6: Dictionary of ODS Language Statements, ODS EXCEL Statement (pp. 292-320). Available at <http://support.sas.com/documentation/cdl/en/odsug/67921/PDF/default/odsug.pdf>

PROC MSCHART:

SAS (2015). *SAS 9.4 Output Delivery System: Procedures Guide*. Chapter 7: (Preproduction) MSCHART Procedure (pp. 177-212). Available at <http://support.sas.com/documentation/cdl/en/odsproc/67922/PDF/default/odsproc.pdf>.

PROC TEMPLATE:

Haworth, Lauren (2006). "SAS with STYLE: Creating your own ODS Style Template for RTF Output." *Proceedings of SUGI 30*. Available at <http://www2.sas.com/proceedings/sugi30/132-30.pdf>.

Zender, Cynthia (2010). "SAS Style Templates: Always in Fashion." *Proceedings of the 2010 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings10/033-2010.pdf>.

TAGATTR, FORMULAS:

Gebhart, Eric (2010). "ODS ExcelXP: Tag Attr Is It! Using and Understanding the TAGATTR= Style Attribute with the ExcelXP Tagset." *Proceedings of the 2010 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings10/031-2010.pdf>

TRAFFIC LIGHTING:

Hadden, Louise (2007). "STOP! WAIT! GO! See What Traffic-Lighting Can Do For You!" *Proceedings of SUGI 30*. Available at <http://www2.sas.com/proceedings/sugi31/142-31.pdf>.

Karp, Andrew (2011). "Traffic-Lighting Your Reports the Easy Way with PROC REPORT and ODS." *Proceedings of the 2010 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings11/290-2011.pdf>.

Other ODS Papers by Gina Huff:

Huff, Gina (2011). "Absolutely Fabulous: Tips on Creating a Publication-Ready Report Using ODS Absolute Layout Functionality." *Proceedings of the 2011 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings11/293-2011.pdf>.

Huff, Gina (2014). "Simple ODS Tips to Get RWI (Really Wonderful Information) from your RWI (Report Writing Interface)." *Proceedings of the 2014 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings14/1689-2014.pdf>.

CONTACT INFORMATION

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

Gina Huff

Western Kentucky University
Office of Institutional Research
1906 College Heights Blvd. #11011
Bowling Green, KY 42101

(270) 745-3250
gina.huff@wku.edu



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.