

Excel with ODS EXCEL Destination and PROC REPORT!

Devi Sekar, RTI International

ABSTRACT

ODS EXCEL destination for creating Microsoft Excel workbooks is available starting with SAS® 9.4M3. This destination is an extremely easy and handy tool for producing ad-hoc as well as production Excel reports. The ODS EXCEL destination has several advantages over ODS ExcelXP tagset. With the ODS EXCEL destination, you can bring all those powerful features available with the REPORT procedure such as predefined styles, traffic-lighting, custom formatting, and compute block flexibility straight into your Excel reports. This paper covers various techniques that you can use with PROC REPORT and the ODS EXCEL destination, to bring excellence to your work.

INTRODUCTION

At RTI International, our team processes U.S. Center for Medicare and Medicaid Services (CMS) Teradata data tables and generates summary tables in SAS. These summary tables are then used to produce a variety of Excel reports for CMS consumption.

The ODS EXCEL destination creates native Microsoft Excel .xlsx files, supports graphic images, and offers many advantages over the ExcelXP tagset. The ODS EXCEL destination works on Linux/Unix operating systems as well and the reports are generated on Linux/Unix. You can quickly create multi-sheet Excel workbooks with this destination.

With the ODS EXCEL destination, each element of the report - titles, footnotes, column width, row height, changes in color or font - is programmed through various options within SAS. Both the formatting and the data placement are done in SAS. Although the initial work of formatting the report may be time consuming, ODS EXCEL provides stable and reproducible output for production reports. It is an excellent path for many of us who are struggling to move the data from SAS to Excel in the quickest possible way.

PROC REPORT features coupled with the ODS EXCEL destination bring excellent results. In this paper, I will cover several examples that use PROC REPORT features and the ODS EXCEL destination options. Although the ODS EXCEL destination supports Graphics procedures, these procedures are not covered in this paper. I hope SAS users will benefit from the techniques presented here.

REQUIREMENTS

To use the techniques described in this paper, you must have the following software:

- Base SAS® 9.4M3 and later, on any supported operating system (ex: Linux, SAS/Grid, Windows) and hardware.
- Microsoft Excel 2007 or later.

LIMITATIONS

You can use the **ODS EXCEL destination** options with all SAS procedures, but ODS style overrides apply only to the PRINT, REPORT, and TABULATE procedures. We will review several ODS EXCEL destination options and style overrides in this paper.

The **ODS EXCEL destination** creates a new workbook on each execution and cannot be used to update an existing workbook. This means, you will not be able to have static worksheets in your workbook and update only certain sheets. You will need to program each sheet in SAS.

WHY PROC REPORT?

PROC REPORT has the flexibility to create simple to complex reports with summary lines, text lines, and conditional formatting. Compute Block in PROC REPORT is very powerful – it can help calculate new columns in the report using the existing columns, add blank lines and text with the LINE statement, use the CALL DEFINE statement within the compute block to apply style attributes such as color and formatting. PRINT and TABULATE procedures have some of these capabilities but not all. With the availability of the ODS EXCEL destination, it is worth investing time in learning PROC REPORT.

HOW TO USE THE ODS EXCEL DESTINATION

ODS (Output Delivery System) is part of Base SAS software that enables you to generate different types of output from your procedure code. ODS destination supports several types of output including but not limited to HTML, RTF, PDF. ODS style controls the appearance of the output.

ODS uses a simple “wrapper method” to delineate what output should be saved to the file. You will initiate Excel at the beginning of the code that will create your output and then close it at the end. Below are the general ODS statements to generate a .xlsx output:

```
ods _all_ close;
ods excel file="/myshare/sample.xlsx" style= [style name]; /*Opens Excel*/
  .. SAS program code here, to generate output
  ..
ods excel close; /*Closes Excel */
```

This is a simple SAS program that generates an Excel report:

```
ods excel file="/myshare/classdata.xlsx";
title "SASHELP.CLASS data";
proc print data=sashelp.class;
run;
ods excel close;
```

APPLYING SAS STYLES TO YOUR WORKBOOK

ODS EXCEL statement option supports several SAS styles. You can customize the look of the worksheet by specifying any one of the SAS styles in the style options.

Example: `ods excel file="/myshare/classdata.xlsx" style=htmlblue`. This example uses the SAS style **htmlblue**.

You can view the list of styles that are available at your site by submitting the statements below:

```
proc template;
  list styles;
run;
quit;
```

ODS EXCEL STATEMENT SUBOPTIONS

ODS EXCEL OPTIONS feature provides a variety of suboptions to control various aspects of your Excel output. There are over 45 suboptions available in SAS 9.4TSM3 and the list is growing with each new release.

Suboptions are specified in an ODS statement using the OPTIONS keyword:

```
ods excel options (option-name1='value1' option-name2='value2' ...);
```

Note that the value that you specify for a suboption remains in effect until the Excel destination is closed or until the suboption is set to another value. Because multiple ODS Excel **options** statements are

allowed within the same SAS program, it is a good practice, in terms of functionality and code readability, to explicitly reset an option to its default value each time you are finished using that specific option.

Let us review some of the most frequently used suboptions with their possible values and their default values for their arguments. A complete list of suboptions is available in ODS EXCEL Statement in SAS® 9.4 Output Delivery System. (*Please see ODS EXCEL reference in the References section.*)

EMBEDDED_TITLES= 'OFF' | 'ON' (alias 'YES'|'NO') – default is OFF

ON displays titles in the worksheet.

OFF does not display titles in the worksheet. They only show up in Page Setup.

EMBEDDED_FOOTNOTES= 'OFF' | 'ON' (alias 'YES'|'NO') – default is OFF

ON displays footnotes in the worksheet.

OFF does not display footnotes in the worksheet. They only show up in Page Setup.

SHEET_NAME=*'text-string'*

specifies the name for the next worksheet. This name is used along with the worksheet counter to create a unique name.

SHEET_LABEL=*'text-string'*|'NONE')

used as the first part of the name in the worksheet label instead of the predefined string.

SHEET_INTERVAL= 'TABLE' | 'NONE' | 'PROC' | 'BYGROUP' | 'PAGE' . Default value is 'TABLE'.

This suboption controls if the report output needs to be on the same sheet or a new sheet. There are five possible values available for this suboption. While going through the example worksheets in this paper, you will understand the usage of each of these choices.

TABLE (alias **Output**) creates a worksheet for each table.

NONE creates one worksheet with multiple outputs.

PROC creates a worksheet of all of the procedure output.

BYGROUP (Alias **BYGROUPS**) creates a new worksheet after each BYGROUP.

PAGE creates a worksheet for each page of procedure output.

START_AT= 'string' specifies a starting cell for the report. The default is to start at column 1 and row 1.

Example: ods excel options(start_at="5,2"). This option cannot be changed in the middle of a sheet.

TAB_COLOR=*'string'* specifies the color for the worksheet tab.

ABSOLUTE_COLUMN_WIDTH =*'number-list '* | 'NONE' specifies the default column widths.

Lists widths to use for columns instead of allowing SAS to determine the column width (measured widths).

The *number-list* is a comma separated list of numbers. You can use 'NONE' to reset to the default.

Example: ods excel options(absolute_column_width='20,10,10') sets the column width for the first column as 20 and for 2nd and 3rd columns as 10. And these widths repeat again for the next set of three columns and so on.

ABSOLUTE_ROW_HEIGHT =*'number_list '* Specifies row heights. Heights repeat for each set of rows.

Example: ods excel options(absolute_row_height='20') sets row_height to 20 for all rows.

ods excel options(absolute_row_height='20,15') sets row heights to 20 and 15 for each set of two rows.

ROW_HEIGHTS =*'number_list '*) specifies the height of the row based on the row type.

The parameters of this option are positional, but not all values must be specified. A value of 0 means that the height should be taken from the style. The first value is the height for table **header** rows. The next is the height for the table **body** rows. The next value is the row height for **BY lines**. The fourth is for **titles**, the fifth is for **footers**, the sixth is the page break height, and the last value is the height for paragraph skip.

Example: ods excel options (row_heights = "50,16,20,30,30")

In this example, header height is set to 50, all row heights in the table are set to 16, BY lines height is set

to 20, title and footnote heights are set to 30.

AUTOFILTER = 'ALL' | 'NONE' | 'range ' turns on filtering for specified columns in the worksheet.

Example: ods excel options(autofilter='1-5') sets auto filter to columns 1 to 5 in the excel sheet.

ZOOM='number' indicates the initial zoom level on the worksheet. Default is 100.

Example: ods excel options(zoom='75') sets zoom level for the worksheet to 75%.

ODS STYLE OVERRIDES

You can customize the columns or rows in the excel report, (for example: change font weight, change text color, change font height) by using the style overrides. Style overrides are supported by the PRINT, REPORT, and TABULATE procedures. When applying style overrides, you will need to specify the location for each style. Style overrides can be specified several ways. Here are the two most common ways of specifying style overrides.

- a. style(location)= [style-attribute-name1=value1 style-attribute-name2=value2 ...]
- b. style(location)=style-element-name

The first format uses individual style attributes defined inline. The second format uses a user defined or SAS style element name.

In PROC REPORT, you can apply style overrides for these five locations: REPORT, HEADER, COLUMN, SUMMARY, and LINES. The REPORT location controls the appearance of the report. The HEADER location controls the appearance of the report header. The COLUMN location controls the appearance of data cells. The SUMMARY location controls the appearance of summary lines. The LINES location controls the appearance of any notes lines.

For example, if you include the following style options in the PROC REPORT statement, they apply to the report, header, all columns in the report, summary lines in the report, and any notes lines in the report:

- style(report)= [cellspacing=5 borderwidth=10 bordercolor=blue]
- style(header)= [font_weight=bold font_size=11pt just=left]
- style(column)= [just=center font_size=10pt]
- style(lines)= [just=left font_weight=bold]
- style(summary)= [background=lightgrey font_size=11pt just=r];

In PROC REPORT code, you can also apply style overrides to the data cells for individual columns in the define statement. For example:

```
define myvar / style(column)= [background=yellow font_size=10pt just=left];
```

USING A COMPUTE BLOCK WITHIN PROC REPORT

Once you start using the ODS EXCEL destination and PROC REPORT, you will quickly realize the need for a compute block. Let us review some of the important concepts related to compute block:

A **compute block** starts with a COMPUTE statement and ends with an ENDCOMP statement. Between these two statements, you can use other SAS statements (ex: assignment, CALL DEFINE, and LINE statements) that customize your output. PROC REPORT processes a data set by reading the variables in the order in which they appear from left to right in the COLUMN statement. The procedure builds the report one column and one row at a time, and COMPUTE statements are executed as the report is built.

You create a compute block with the COMPUTE statement. A COMPUTE statement can contain several arguments of the following types:

report-item - a data set variable, a statistic, or a computed variable.

location - specifies at what point in the process the compute block executes in relation to the value for target. The location can be at the top or bottom of the report; before or after a set of observations.

target - controls when the compute block executes. You can specify a target if you specify a location (BEFORE or AFTER) for the COMPUTE statement. A target can be a group or order variable or a `_PAGE_` variable.

A compute block that is associated with a report item can perform the following tasks:

- define a variable that appears in a column of the report but is not in the input data set.
- within the compute block, you can use most SAS language elements to perform calculations.
- change row attributes or column attributes within the report using a CALL DEFINE statement.
- define or change the value for a report item, such as showing the word “Total” on a summary line.
- Include text lines or blank lines to the report.
- A compute block that is associated with a location can write a customized summary.
- A PROC REPORT step can contain multiple compute blocks, but they cannot be nested.

In general, compute blocks are executed in the following order:

1. COMPUTE report-item;
2. COMPUTE BEFORE;
3. COMPUTE BEFORE target;
4. COMPUTE BEFORE `_PAGE_`;
5. COMPUTE AFTER;
6. COMPUTE AFTER target;
7. COMPUTE AFTER `_PAGE_`;

For details about all the arguments that are available in the COMPUTE statement, see the Base SAS 9.x Procedures Guide for your SAS release(support.sas.com/documentation/onlinedoc/base/index.html). For an in-depth look at using a Compute Block, refer to the excellent papers on Compute Block (*Eslinger 2015 and Zender 2017*).

EXAMPLE EXCEL WORKBOOK

In this paper, we will go through the SAS code used to generate a list of five Reports in an Excel workbook. These reports were generated using the ODS EXCEL destination and PROC REPORT. All the five reports were generated using SASHELP.CARS dataset. The five worksheets include:

- a. TOC worksheet - Table of Contents -with the ability to navigate to each worksheet.
- b. Report1a - List of Cars By Make, Type and Model with (profit pre-computed)
- c. Report1b - List of Cars By Make, Type and Model (with profit computed using an Excel formula)
- d. Report2 - Average MPG City, MPG Highway, MSRP, and Invoice Price by Type, Make, and Origin
- e. Report3 - Average invoice price and average MSRP By Make and across Type within Origin.

A. CREATE TABLE OF CONTENTS (TOC) WORKSHEET

Below is a screenshot of the TOC worksheet.

Table	Description
Report1a	List of Cars By Make, Type and Model (with profit pre-computed)
Report1b	List of Cars By Make, Type and Model (with profit computed using an Excel formula)
Report2	Average MPG City, MPG Highway, MSRP, and Invoice Price by Type, Make, and Origin
Report3	Average MSRP and Invoice Price By Type and across Make within Origin

solid border lines with style(header) options in &header_col_style macro variable

Clickable hyperlinks

Figure 1 : Screenshot of TOC worksheet

Please note that the cells in column A are hyperlinks. For example, when you click on Report 1a, it takes you to the Report 1a. Below is the SAS program that generate the Table of Contents Spreadsheet:

/ Create a dataset with two columns – worksheet name and description*/*

```
data toc;
    length tab_name $10 tab_description $100;
    tab_name = "Report1a";
    tab_description = " List of Cars By Make, Type and Model (with profit
pre-computed) ";
    output;
    tab_name = "Report1b";
    tab_description = " List of Cars By Make, Type and Model (with profit
computed using Excel formulas)";
    output;
    tab_name = "Report2";
    tab_description = " Average MPG City, MPG Highway, MSRP, and Invoice
Price by Type, Make, and Origin ";
    output;
    tab_name = "Report3";
    tab_description = " Average MSRP and Invoice Price By Type and across
Make within Origin";
    output;
run;
```

/ SAS Program to generate the TOC Excel worksheet. */*

/ ODS Excel statement below */*

① ods excel file= '/mydir/SESUG2017/SESUG2017_Tables.xlsx' style=pearl

```

options (embedded_titles = 'yes' embedded_footnotes = 'yes');

❷ %let header_col_style = style(header)= [font_weight=bold font_size=11pt
color=black bordertopstyle=solid bordertopwidth=0.1pt bordertopcolor=black
borderbottomstyle=solid
borderbottomwidth=0.1pt borderbottomcolor=black just=center]
style(column)=[font_size=11pt];

/* Excel sheet options for TOC work sheet */

❸ ods excel options (sheet_name = 'TOC'
sheet_interval = 'Table'
absolute_column_width='12,100'
row_heights='30,20,20,20,20');

❹ title j=left bold "List of Tables";
proc report data=toc
&header_col_style;
column tab_name tab_description;
define tab_name / display format= $15. left "Table"
style(column) = [textdecoration=underline
foreground=blue];
define tab_description / display format= $100. left "Report
Description"
style(column) = [foreground=blue];

compute tab_name;
urlstring = "#" || strip(tab_name) || "!A1";
call define (_col_, 'url', urlstring);
endcomp;

run;

```

Program Description

- ❶ Open the ODS destination for Excel and specify options that apply to every sheet in this workbook:
 - a. **embedded_titles** and **embedded_footnotes** options allow the title and footnotes to display directly on the worksheet. The default is OFF. If you do not specify these options, the title and footnote do not show up on your worksheet. They will only show up in Page Setup.
 - b. **Style=pearl** option is one of the SAS styles available on my SAS system.
- ❷ **&header_col_style** macro variable lists the style options I will be using for the header row and data cells for every worksheet in this workbook. Style(header) options include options to set top and bottom borders for the header row in the Excel report, font_size to 11, bold the text, and center header columns text. Style(column) option sets the font size for the data cells in each column to 11. It is a good idea to include all those options that apply to majority of the columns and header text at this place. You can override the style attributes for individual columns in the define statements.
- ❸ ODS EXCEL options statement lists the options specific to the TOC worksheet. SHEET_NAME is given the value TOC. Column widths are specified for the two columns. Row heights are specified for header, body, bylines, title and footnote. SHEET_INTERVAL is given the option "Table". The default behavior for excel destination is to place each table in its own worksheet. By default, PROC REPORT creates just one table, so just one sheet is created.
- ❹ PROC REPORT procedure code
 - a. Title statement- **j=left** option left justifies the title.

- b. Specify the report statement with options. By default, PROC REPORT runs without the REPORT window and sends its output to the open output destinations. Style options for header and columns are listed in &header_col_style macro variable
- c. Specify the report columns and define style attributes for individual columns with the define statements.

```
column tab_name tab_description;

define tab_name / display format= $15. left "Table" style(column) = [textdecoration=underline foreground=blue];

define tab_description / display format= $100. left "Report Description" style(column) = [foreground=blue];
```

- d. compute statement block for the tab_name – These two statements make each cell a clickable hyperlink in the Table column. For example, urlstring is set to “#Report1a! A1” for first observation. The **call define** statement makes the first column into hyperlinks. Clicking on the cell A4 takes you to the cell A1 in Report1a worksheet.

```
compute tab_name;
  urlstring = "#" || strip(tab_name) || "! A1";
  call define (_col_, 'url', urlstring);
endcomp;
```

B. REPORT1A WORKSHEET

This sheet generates list of cars by Make, Type and Model with their Invoice Price, MSRP, Profit and Profit Percentage.

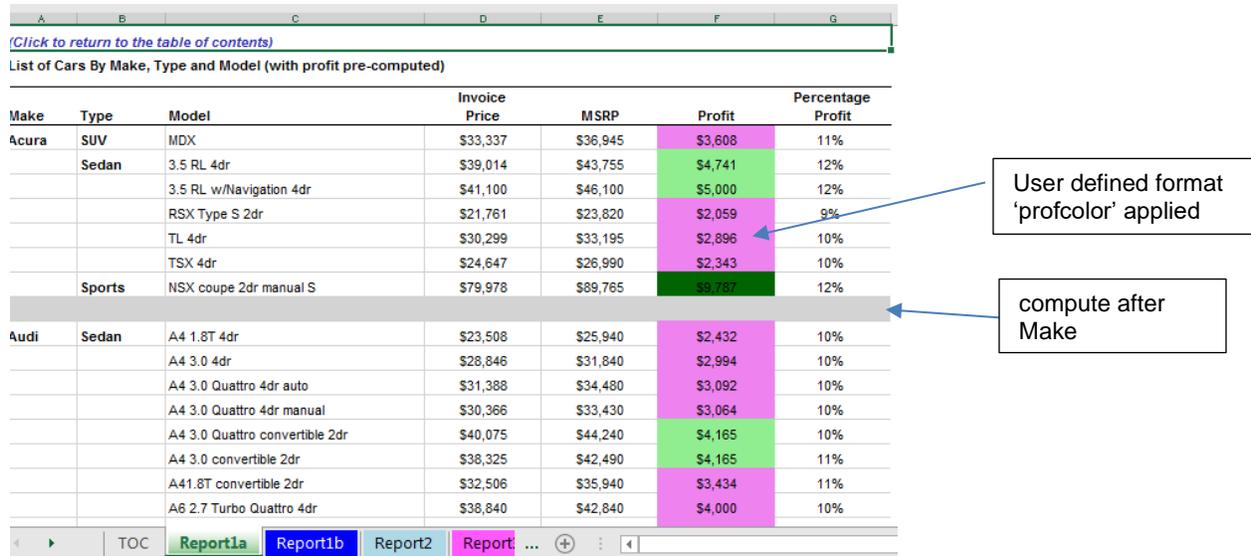


Figure 2: Screenshot of the Report1a Worksheet

Below is the SAS program to generate the above worksheet:

/* Create color format to color code different profit ranges */

```
1 proc format;
  value profcolor
    low - 0 = "red"
    0.01 - 2000 = "lightblue"
```

```

2000.01 - 4000 = "violet"
4000.01 - 5000 = "lightgreen"
5000.01 - high = "darkgreen";
run;

/* Excel sheet options for Report1a worksheet */

❷ ods excel options (sheet_name='Report1a'
                    absolute_column_width='12,15,45,20,20,20,20'
                    row_heights='30,16,20,20,20'
                    sheet_interval="Table"
                    tab_color='green');

title1 j=left bold italic underlin=1 color=bib link="#"'TOC'!A2" '(Click to
return to the table of contents)';

title2 j=left "List of Cars By Make, Type and Model (with profit pre-
computed) ";

❸ proc report data=sashelp.cars split="*"
    &header_col_style;
    column make type model invoice msrp profit profit_pct;
    define make/order "Make" style(header)=[just=left] style(column) =
[font_weight=bold];
    define type /order "Type" style(header)=[just=left] style(column) =
[font_weight=bold];
    define model/order "Model" style(header)=[just=left];
    define msrp/format=dollar12. "MSRP" center;
    define invoice/format=dollar12. "Invoice*Price" center;
    define profit /computed "Profit" format=dollar12. center
style=[background=profcolr.];

/* See notes below for displaying percentages with decimals*/

    define profit_pct /computed "Percentage*Profit" format=percent8.2
center;

❹ compute profit;
    profit = msrp.sum-invoice.sum;
endcomp;
compute profit_pct;
    profit_pct = profit/invoice.sum;
endcomp;

❺ compute after make/style=[background=lightgrey];
    line ' ';
endcomp;
run;

```

Program Description

❶ Create a format **profcolor.** with different color values for profit ranges to be used to color code the profit column in the Excel report.

② Specify ODS EXCEL options statement for this worksheet. Please note that your ODS EXCEL destination is still open. The options you specify here will apply to this worksheet. Please note that options specified in the beginning with the ODS EXCEL file statement above are still available to this worksheet. In the **absolute_column_width** option, the column width for each of the seven columns in the report is specified as '12,15,45,20,20,20,20'. The **row_heights** option is specified as '30,16,20,20,20', row_height for header is 30, row_height for body lines is 16, row_height for bylines, title and footnotes is also set to 20. The **Sheet_interval** option is set to "Table", since it is just one table generated from Proc Report. Tab_color option is set to Blue.

③ PROC REPORT procedure code.

- a. Title statement- j=l option left justifies the title.
- b. Specify the report statement with options. Split="*" option is used to split the text in column header labels. Style options for header and columns are listed in the &header_col_style macro variable.

Specify the report columns and define the attributes for each column with the define statements.

```
column make type model invoice msrp profit profit_pct;
define make/order "Make" style(header)=[just=left] style(column) = [font_weight=bold];
define type /order "Type" style(header)=[just=left] style(column) = [font_weight=bold];
define model/order "Model" style(header)=[just=left];
define msrp/format=dollar12. "MSRP" center;
define invoice/format=dollar12. "Invoice*Price" center;
define profit /computed "Profit" format=dollar12. center style=[background=profcolr.];
define profit_pct /computed "Percentage*Profit" format=percent8.2 center;
```

The option **just=center** is specified in the **&header_col_style** macro variable. This option applies to all the headers for all columns. You can override this option for specific columns by applying **style(header)=[just=left]** option in the define statement. In the code above, the column header for **Make**, **Type**, and **model** columns are left justified. **Style(column)=[font_weight=bold]** is applied to the data cells for **Make** and **Type** columns.

Profit and **profit_pct** are computed variables, created using a compute statement. **profcolor** format is used in the **style=[background=profcolr.]** option for the **profit** variable. This technique sets the background color for the **profit** column, based on the color range for **profit**.

④ compute statements code to create **profit** and **profit_pct** variables.

```
compute profit;
    profit = msrp.sum-invoice.sum;
endcomp;
compute profit_pct;
    profit_pct = profit/invoice.sum;
endcomp;
```

By default, all numeric variables are analysis variables, and the default statistic is SUM. Even if you do not include a statistical keyword in a DEFINE statement, you must use the compound name in the compute block. For example, if you use msrp instead of msrp.sum in the above code, you will see the following note in the log:

NOTE: Variable msrp is uninitialized.

Please note that even though the **percent8.2** format is used for the **profit_pct** variable, decimal values for percentages are not showing in the report. This is a glitch in SAS9.4M3 release. We can overcome this problem by using a **tagattr** style override for the percentage format. Use the statements below to display percentages with decimal values for **profit_pct** variable.

```
define profit_pct /computed "Percentage*Profit" style(column)=[tagattr='format: #,##0.0\%']
center;
```

```
compute profit_pct;
    profit_pct = (profit/invoice.sum) *100;
endcomp;
```

In the second compute statement, the variable profit is used without specifying it as a compound name, since it is a computed variable.

- 5 This compute statement code inserts a light grey line after each **Make** value changes.

```
compute after make/style=[background=lightgrey];
line ' ';
endcomp;
```

C. REPORT1B WORKSHEET

This worksheet is exactly same as Report1a, except that the profit value is calculated in the Excel sheet using Excel formulas, instead of pre-populating this column! When you hover over the values in this column, you can see the formula associated with it.

Make	Type	Model	Invoice Price	MSRP	Profit
Acura	SUV	MDX	\$33,337	\$36,945	\$3,608
	Sedan	RSX Type S 2dr	\$21,761	\$23,820	\$2,059
		TSX 4dr	\$24,647	\$26,990	\$2,343
		TL 4dr	\$30,299	\$33,195	\$2,896
		3.5 RL 4dr	\$39,014	\$43,755	\$4,741
		3.5 RL w/Navigation 4dr	\$41,100	\$46,100	\$5,000
	Sports	NSX coupe 2dr manual S	\$79,978	\$89,765	\$9,787
Audi	Sedan	A4 1.8T 4dr	\$23,508	\$25,940	\$2,432
		A4 1.8T convertible 2dr	\$32,506	\$35,940	\$3,434
		A4 3.0 4dr	\$28,846	\$31,840	\$2,994
		A4 3.0 Quattro 4dr manual	\$30,366	\$33,430	\$3,064
		A4 3.0 Quattro 4dr auto	\$31,388	\$34,480	\$3,092

Figure 3 - Screenshot of the Repor1b Worksheet

Below is the SAS Program that generated the above excel sheet - Report1b.

```
/* Excel sheet options for Report1b worksheet */

ods excel options (sheet_name = 'Report1b'
                    absolute_column_width='12,15,45,20,20,20'
                    row_heights='30,16,20,20,20'
                    sheet_interval="Table"
                    tab_color='blue');
```

```
title1 j=left bold italic underlin=1 color=bib link="#'TOC'!A2" '(Click to return to the table of contents)';
```

```
title2 j=left " List of Cars By Make, Type and Model (with profit computed using Excel formulas)";
```

```
proc report data=sashelp.cars split="*"
  &header_col_style;
  column make type model invoice msrp profit;
  define make/order "Make" style(header)=[just=left] style(column) =
[font_weight=bold];
  define type/order "Type" style(header)=[just=left] style(column) =
[font_weight=bold];
  define model/order "Model" style(header)=[just=left];
  define msrp/format=dollar12. center "MSRP";
  define invoice/format=dollar12. center "Invoice*Price";
  ❶ define profit /computed "Profit" center format=dollar12.
    style= {tagattr= 'formula:RC[-1]-RC[-2]'};

  ❷ compute profit;
    profit = 0;
  endcomp;

  compute after make/style=[background=lightgrey];
    line ' ';
  endcomp;
run;
```

Program Description

- ❶ The SAS program for Report1b is same as the code for Report1a, except that it is using an Excel formula with **tagattr style** in the style options in the define statement for **profit**. When you hover over the profit values in the Excel report, you can view the formula for **profit!**
- ❷ Please note that Excel must read the **profit** column as numeric and this column cannot be missing for the Excel formula to work. This is accomplished by setting the **profit** value to zero with a compute statement.

D. REPORT2 WORKSHEET

In this worksheet, we will review more features available in PROC REPORT and how to place more than one table in the same worksheet.

Included is a screenshot of the two reports in Report2 Worksheet

Type	Make	Origin	MPG City Mean	MPG Highway Mean	MSRP Mean	Invoice Price Mean
Hybrid	Honda	Asia	53.00	58.50	\$19,625	\$18,181
	Toyota	Asia	59.00	51.00	\$20,510	\$18,926
Hybrid	Mean:		55.00	56.00	\$19,920	\$18,429
SUV	Acura	Asia	17.00	23.00	\$36,945	\$33,337
	BMW	Europe	16.00	22.50	\$44,598	\$40,797
	Suzuki	Asia	18.50	22.00	\$20,431	\$19,628
	Toyota	Asia	17.00	21.20	\$33,278	\$29,616
	Volkswagen	Europe	15.00	20.00	\$35,515	\$32,243
	Volvo	Europe	15.00	20.00	\$41,250	\$38,851
SUV	Mean:		16.10	20.50	\$34,790	\$31,625
Sedan	Acura	Asia	20.40	27.20	\$34,772	\$31,364
	Audi	Europe	18.62	25.92	\$41,108	\$37,401
	BMW	Europe	19.23	28.08	\$43,380	\$39,699
	Buick	USA	19.43	29.00	\$30,057	\$27,465
	Cadillac	USA	18.00	25.75	\$43,708	\$40,107

break after type/summarize

color code type summary line with CALL DEFINE in compute after type

Figure 4 - First Table in the Report2 Worksheet

The second report is a summary report at the bottom of the sheet in the Report2 worksheet:

Type	MPG City Mean	MPG Highway	MSRP Mean	Invoice Price Mean
Hybrid	55.00	56.00	\$19,920	\$18,429
SUV	16.10	20.50	\$34,790	\$31,625
Sedan	21.08	28.63	\$29,774	\$27,390
Sports	18.41	25.49	\$53,387	\$48,473
Truck	16.50	21.00	\$24,941	\$22,617
Wagon	21.10	27.90	\$28,841	\$26,646

solid border line with compute after

Figure 5 - Second Table in the Report2 Worksheet

Below is the SAS program to generate the above worksheet:

```
/* Excel sheet options for the Report2 worksheet */
```

```

❶ ods excel options (sheet_name ='Report2'
                    sheet_interval="None"
                    absolute_column_width='15,20,20,20,20,20,20'
                    row_heights='30,16,20,20,20'
                    tab_color='lightblue');

title;
title1 j=left bold italic underlin=1 color=bib link="#"'TOC'!A2" '(Click to
return to the table of contents)';
title2 j=left "Average MPG City, MPG Highway, MSRP, and Invoice Price by
Type, Make, and Origin";

❷ proc report data=sashelp.cars split="*"
    &header_col_style;
    column type make origin MPG_City MPG_Highway msrp Invoice;

    define type /group "Type" style(header)=[just=left] style(column) =
[font_weight=bold];
    define make/order "Make" style(header)=[just=left];
    define origin/order "Origin" style(header)=[just=left];
    define MPG_City/analysis mean format=8.2 "MPG City*Mean" center;
    define MPG_Highway/analysis mean format=8.2 "MPG Highway*Mean"
center;
    define msrp/ analysis mean format=dollar12. center "MSRP*Mean";
    define invoice/ analysis mean format=dollar12. center "Invoice
Price*Mean";

❸ break after type/summarize;

❹ compute after type;
    make = "Mean:";
    if type = "Hybrid" then
        call define (_row_, 'style', 'style=[background=lightgreen]');
    else if type in ('SUV','Truck') then
        call define (_row_, 'style', 'style=[background=lightred]');
    else
        call define (_row_, 'style', 'style=[background=yellow]');
    endcomp;

❺ compute after/style= [bordertopstyle=solid bordertopwidth=0.1pt
bordertopcolor=black];
    line ' ';
    endcomp;
run;

/* Code for the second report */

title;
title1 j=left bold italic underlin=1 color=bib link="#"'TOC'!A2" '(Click to
return to the table of contents)';
title2 j=left "Average MPG City, MPG Highway, MSRP, and Invoice Price by
Type";

proc report data=sashelp.cars split="*"
    &header_col_style;
    column type MPG_City MPG_Highway msrp Invoice;

```

```

define type /group "Type" style(header)=[just=left] style(column) =
[font_weight=bold];

define MPG_City/analysis mean format=8.2 "MPG City*Mean" center;
define MPG_Highway/analysis mean format=8.2 "MPG Highway*Mean" center;
define msrp/ mean format=dollar12. center "MSRP*Mean";
define invoice/ mean format=dollar12. center "Invoice Price*Mean";

compute after/style= [bordertopstyle=solid bordertopwidth=0.1pt
bordertopcolor=black];
    line ' ';
endcomp;

run;

```

Program Description

- ① Please note that SHEET_INTERVAL is set to “None” in the ODS EXCEL options statement. **SHEET_INTERVAL = ‘None’** places multiple tables in the same worksheet. This sheet has two tables from two PROC REPORT outputs.
- ② PROC REPORT procedure code. In this code, the **mean** statistic is used in the define statements for all analysis variables. Ex:

```
define MPG_City/analysis mean format=8.2 "MPG City*Mean" center;
```
- ③ A summary of analysis variables is displayed in the reports after each **Type**, with the statement below. In this case, you will see the **mean** values for all four analysis variables for each **Type**.

```
break after type/summarize;
```
- ④ The **compute block** statement used to color code the summary rows based on the **Type** by using **compute after type** and CALL DEFINE statements. The variable **Make** is set to the value ‘Mean’. The **call define (_row_, 'style', 'style=[background=xxxxxx]');** statement with different background colors for different Type value colors the Type row in the Excel report.
- ⑤ This compute block inserts a line after the last row and makes the top border solid by using **compute after/ style= [bordertopstyle=solid bordertopwidth=0.1pt bordertopcolor=black]**.

E. REPORT3 WORKSHEET

In this example, you will see how to display the metrics across for a categorical variable in PROC REPORT without first transposing the data. Also shown is how to generate one report per BY group value. For in-depth understanding of the **Across** option, please see Zender 2014 in the references section.

Included is a screenshot of the Report3 Worksheet

Report3 : Average MSRP and Invoice Price By Type and across Make (Origin: Asia)												
	Acura		Honda		Hyundai		Infiniti		Isuzu		Kia	
Type	Average MSRP	Average Invoice Price										
Hybrid			\$19,625	\$18,181								
SUV	\$36,945	\$33,337	\$22,037	\$20,199	\$21,589	\$20,201			\$26,149	\$24,619	\$19,635	\$18,63
Sedan	\$34,772	\$31,364	\$20,525	\$18,618	\$16,939	\$15,512	\$36,212	\$33,027			\$15,899	\$14,96
Sports	\$89,765	\$79,978	\$33,260	\$29,965	\$18,739	\$17,101						
Truck												
Wagon							\$35,645	\$32,439			\$11,905	\$11,41

Report3 : Average MSRP and Invoice Price By Type and across Make (Origin: Europe)												
	Audi		BMW		Jaguar		Land Rover		MINI		Mercedes-Benz	
Type	Average MSRP	Average Invoice Price	Average MSRP	Average Invoice Price								
SUV			\$44,598	\$40,797			\$45,832	\$41,851			\$61,670	\$57,40
Sedan	\$41,108	\$37,401	\$43,380	\$39,699	\$53,123	\$48,402			\$18,499	\$16,787	\$54,573	\$50,77

across option for make variable

One report per By Group value

Figure 6 - Screenshot of the Report3 Worksheet

Below is the SAS program to generate this worksheet:

```
/* Excel sheet options for Report3 worksheet */
```

```
ods excel options (sheet_name='Report3'
                  sheet_interval="Proc"
                  absolute_column_width='16'
                  row_heights='65,16,20,20,20'
                  tab_color='lightpurple'
                  frozen_rowheaders='1'
                  column_repeat='1'
                  title_footnote_nobreak='yes'
                  scale='70'
                  orientation='landscape');

title;
title1 j=left bold italic underlin=1 color=bib link="#'TOC'!A2" '(Click to
return to the table of contents)';
title2 j=left bold height=11pt "Report3: Average MSRP and Invoice Price By
Type and across Make (Origin: #Byval(Origin)) ";

proc sort data=sashelp.cars out=cars;
  by origin;
run;

options nobyline missing=" ";

proc report data=cars split="*"
  &header_col_style;
  by origin;
  column type make, (msrp invoice);
  define type/group "Type" style(header)=[just=1] style(column) =
[font_weight=bold];
  define make/across "";
  define msrp/ mean analysis format=dollar12. center "Average*MSRP";
```

```

define invoice/ mean analysis format=dollar12. center
"Average*Invoice*Price";

compute after/style= [bordertopstyle=solid bordertopwidth=0.1pt
bordertopcolor=black];
  line ' ';
endcomp;
run;

title;
options byline;

ods excel close;

```

Program Description

① In this example, SHEET_INTERVAL is set to “Proc” in the ODS EXCEL options statement. SHEET_INTERVAL = ‘Proc’ places all the tables generated by one PROC REPORT statement in the same worksheet. In this example three tables (for three origins) are generated by PROC REPORT. All three tables are placed on the same sheet with the SHEET_INTERVAL = “Proc” option. With absolute_column_width set to ‘16’, every report column will have the width of 16. The frozen_rowheaders = ‘1’ option freezes the first column, as you scroll through to the right.

② The **By origin** statement generates three tables, one for each **origin** value.

③ In the columns statement, a **comma “,”** followed by the list of variables in parenthesis, transposes the listed variables in parenthesis by the group variable. You will need to specify the **“across”** option in the **define statement** for the variable across which you want to transpose the data.

In the code above, **column type make, (msrp invoice)** statement and **across** option in the **define statement** for **Make** variable transposes the data for the two metrics: **MSRP** and **invoice** for distinct Make values by **Type** as in **Figure 6**. Note that **type** is defined as a **‘group’** variable.

④ Finally close the Excel report by using the **ODS EXCEL close** statement.

CONCLUSION

ODS EXCEL destination is now my preferred method for moving data from SAS to Excel. With the ODS EXCEL destination, we can incorporate all those wonderful, familiar style element capabilities that are available with PDF and RTF destinations. Although the initial work of formatting the report may be time-consuming and at times tedious, it is well worth the effort for creating flawless, high class, and effective production Excel reports.

With the ODS EXCEL destination availability, one can truly make use of the capabilities of PROC REPORT. ODS EXCEL and PROC REPORT make a powerful combination. In this paper, I tried cover some of the PROC REPORT features and the ODS EXCEL destination functionality. I hope the users will get to use the techniques presented in this paper and generate excellent reports.

REFERENCES

ODS EXCEL

ODS EXCEL Statement in SAS® 9.4 Output Delivery System: User's Guide, Fifth Edition

<http://support.sas.com/documentation/cdl/en/odsug/69832/HTML/default/p09n5pw9ol0897n1qe04zeur27rv.htm>

Huff, Gina(2016) "An Excellent Journey: Exploring the New ODS EXCEL Statement." *Proceedings of the 2016 SAS Global Forum Conference*. Available at

<http://support.sas.com/resources/papers/proceedings16/2780-2016.pdf>

Eslinger, Jane (2017) The REPORT Procedure and ODS Destination for Microsoft Excel: The Smarter, Faster Way to Create First-Rate Excel Report. *Proceedings of the 2017 SAS Global Forum Conference*. Available at <http://support.sas.com/resources/papers/proceedings17/SAS0235-2017.pdf>

Delgobbo, Vincent (2017) New for SAS® 9.4: Including Text and Graphics in Your Microsoft Excel Workbooks, Part 2. *Proceedings of the 2017 SAS Global Forum Conference*

<http://support.sas.com/resources/papers/proceedings17/SAS0127-2017.pdf>

Parker, Chevell (2016)"A Ringside Seat: The ODS Excel Destination versus the ODS ExcelXP Tagset." *Proceedings of the 2016 SAS Global Forum Conference*. Available at

<http://support.sas.com/resources/papers/proceedings16/SAS5642-2016.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>

PROC REPORT

REPORT Procedure, Base SAS® 9.4 Procedures Guide, Seventh Edition

<http://support.sas.com/documentation/cdl/en/proc/70377/HTML/default/n0tnmdrpkks51vn0zw7o5hy3fzkq.htm>

COMPUTE STATEMENT

Eslinger, Jane (2015). "The REPORT Procedure: A Primer for the Compute Block." *Proceedings of the SAS Global Forum 2015 Conference*. Available at

<http://support.sas.com/resources/papers/proceedings15/SAS1642-2015.pdf>.

Zender, Cynthia (2017) "Go Ahead and _BREAK_-down: Advanced COMPUTE Block Examples" *Proceedings of the SAS Global Forum 2015 Conference*. Available at

<http://support.sas.com/resources/papers/proceedings17/SAS0431-2017.pdf>

ACROSS STATEMENT

Zender, Cynthia (2014) "Sailing Over the ACROSS Hurdle in PROC REPORT" *Proceedings of the SAS Global Forum 2014 Conference*. Available at

<http://support.sas.com/resources/papers/proceedings14/SAS388-2014.pdf>

CONTACT INFORMATION

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

Devi Sekar
Center for Health Data Analytics
RTI International
3040 East Cornwallis Road
Research Triangle Park, NC 27709
Email: dsekar@rti.org

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.