WUSS Paper 109-2022

Using the Output Delivery System to Create and Customize Excel Workbooks

Joshua M. Horstman, Nested Loop Consulting

ABSTRACT

In years past, SAS® output was limited to the text-based SAS listing. However, the Output Delivery System (ODS) greatly enhanced the capabilities of the SAS system by allowing users to create highly-customizable output in a variety of document formats, including Microsoft Excel® workbooks. This paper provides a brief overview of how to use the ODS EXCEL destination to create excel workbooks and how to customize the various visual attributes of the output such as fonts, colors, styles, and much more.

OVERVIEW OF THE OUTPUT DELIVERY SYSTEM

Before SAS introduced the Output Delivery System, the traditional SAS listing was the primary mechanism by which the SAS system displayed output to the user. While this arrangement served its purpose, it had several limitations. Since it was designed at a time when line printers were widely used, it used a monospace typeface in which characters could be neatly arranged in rows and columns. It did not allow for the use of colors, fonts, graphics, and other visual elements commonly used today. The listing output could also be difficult to import into other applications.

To overcome these limitations and facilitate the creation of publication-quality output, SAS developed the Output Delivery System (ODS). It has been part of the base SAS product since version 7. No separate license is required to access this functionality.

ODS provides extensive flexibility for customizing SAS output:

- Output can be created in a variety of common file formats (including Microsoft Excel files).
- Colors, fonts, borders, shading, and other visual elements can be incorporated.
- Graphics, images, links, and tables can be embedded.
- Pagination can be precisely controlled.
- Accessible output can be generated that will work with assistive technology

In order to allow output to be created in many different file formats, ODS introduces the concept of a "destination." An ODS destination is simply a representation of a location and format to which output can be routed. Some commonly used ODS destinations include RTF, HTML, PDF, POWERPOINT, and EXCEL. Traditional SAS listing output is also still available through the LISTING destination. This paper will focus exclusively on the use of the ODS EXCEL destination.

USING THE ODS EXCEL DESTINATION

The ODS EXCEL destination creates native XLSX files, just like the files created by modern versions of Microsoft Excel. To create such a file using SAS, simply include a pair of ODS EXCEL statements (sometimes referred to as the "ODS sandwich") around the SAS code which generates the desired output. The first ODS EXCEL statement opens the EXCEL destination and directs the output to a specific file location using the FILE= option, while the second statement closes the destination using the CLOSE option. For example, the following sequence of statements would be used to direct the output from the PRINT procedure to an external file.

```
ods excel file="c:\example.xlsx";
   <PROC PRINT code goes here...>;
ods excel close;
```

THE ODS EXCEL OPTIONS OPTION

The ODS EXCEL statement has an OPTIONS option that allows the user to specify a variety of Excel-specific sub-options. One or more pairs of sub-options and values can be enclosed in parentheses after the OPTIONS keyword as follows:

```
ods excel file="c:\example.xlsx"
      options(suboption1=value1 suboption2=value2);
```

The next several examples demonstrate the use of some of the sub-options available using the OPTIONS option. An exhaustive list of the available sub-options can be found in *SAS® 9.4 Output Delivery System: User's Guide, Fifth Edition* (2016).

SPECIFYING THE WORKSHEET NAME

An Excel workbook consists of one or more worksheets, each of which has a name. By default, the worksheets in an Excel workbook created using the ODS EXCEL destination will be named automatically by SAS. However, if a particular worksheet name is desired, it can be specified using the SHEET_NAME sub-option of the OPTIONS option.

```
ods excel file="c:\example.xlsx"
      options(sheet_name="My Output");
   <PROC PRINT code goes here...>;
ods excel close;
```

CREATING MULTIPLE WORKSHEETS WITH BY GROUPS

Many SAS procedures support BY group processing using the BY statement. This is conceptually similar to running the procedure separately for each BY group. When BY group processing is used in conjunction with the ODS EXCEL destination, a separate worksheet is created within the Excel workbook for each unique value of the BY variable (or each unique combination of values when there are multiple BY variables).

The naming of the individual worksheet tabs generated during BY group processing can be controlled using the #BYVALn syntax within the SHEET_NAME sub-option described in the previous example. For example, if #BYVAL1 were used, it would be replaced in each worksheet name with the value of the first BY variable. Similarly, #BYVAL2 would be replaced with the value of the second BY variable, if any.

```
ods excel file="c:\example.xlsx"
      options(sheet_name="#byval1");
   <PROC PRINT code goes here...>;
ods excel close;
```

The value(s) of the BY variable(s) will also appear at the top of each worksheet as a header line in the first row of the spreadsheet. To suppress this header, invoke the NOBYLINE option on the global OPTIONS statement (not to be confused with the OPTIONS option on the ODS EXCEL statement).

```
options nobyline;
```

SPECIFYING A START LOCATION

By default, any SAS output created using the ODS EXCEL destination will begin in the upper-left corner of the Excel worksheet, which Excel designates as cell A1. To place the output in a different location in the worksheet, use the START_AT sub-option in the OPTIONS option to specify a different cell. The following code would start the output in cell C3 instead.

```
ods excel file="c:\example.xlsx" options(start_at="C3");
   <PROC PRINT code goes here...>;
ods excel close;
```

ADJUSTING COLUMN WIDTHS AND ROW HEIGHTS

ODS automatically determines the width for each column in the Excel worksheet based on the lengths of the values and/or labels being placed in that column. You can manually specify the column widths in a couple of different ways. If you wish to specify different widths for different columns, this can be accomplished using in-line styling, which is discussed later in this paper.

However, if you simply wish to specify a common width for all columns in the workbook, this can be done with the ABSOLUTE_COLUMN_WIDTH sub-option on the OPTIONS option. The width can be specified using any ODS-supported unit (cm, mm, in, px, etc.). If you omit the units, the width value will translate directly into Excel's default width units which represent a specific number of characters in the default font.

The following ODS EXCEL statement sets the column width to 3 inches.

```
ods excel file="c:\example.xlsx"
      options(absolute_column_width="3in");
```

Similarly, to change the height of all rows in an Excel worksheet, use the ABSOLUTE_ROW_HEIGHT sub-option.

```
ods excel file="c:\example.xlsx"
      options(absolute_row_height="30");
```

If you wish to specify a height for the header row that is different from the rest of the rows, use the ROW_HEIGHTS sub-option and provide the header row height followed by a comma and then the body row height.

```
ods excel file="c:\example.xlsx"
      options(row_heights="30,15");
```

HIDING OR FREEZING ROWS AND COLUMNS

The extensive functionality provided by Microsoft Excel includes the ability to hide individual or multiple rows and/or columns as well as to freeze header rows and/or columns. Hidden rows or columns still exist in the spreadsheet but simply are not visible until unhidden. Frozen header rows or columns are locked in place and always remain visible as the rest of the spreadsheet is scrolled in any direction.

Both features can be accessed using the ODS EXCEL statement. To hide specific rows and/or columns, use the HIDDEN_ROWS and/or HIDDEN_COLUMNS sub-options to specify the row or column number of the row or column to be hidden. To hide multiple rows or columns, list all the numbers separated by commas. To freeze header rows or columns, use the FROZEN_ROWHEADERS and/or FROZEN_HEADERS sub-options to specify the number of rows and/or columns to be frozen. For example, if the FROZEN_HEADERS sub-option is given a value of "1" then only the top row will be frozen whereas a value of "3" would result in the first three rows all being frozen.

The following example creates an Excel worksheet having the first two rows and the first column (column A) frozen and the 5th and 7th rows and 6th and 8th columns (that is, columns F and H) hidden.

```
ods excel file="c:\example.xlsx"
     options(
          frozen_headers="1"
          frozen_rowheaders="2"
          hidden_rows="5,7"
          hidden_columns="6,8"
     );
```

REMOVING OR MODIFYING CELL BORDERS

Another feature provided by the ODS EXCEL destination is the ability to remove or modify the Excel worksheet cell borders. Unfortunately, this is not available as a sub-option on the OPTIONS option. Rather, we must create a custom style using the TEMPLATE procedure and then apply that style on the ODS EXCEL statement. Custom styles are discussed in more detail later in this paper.

To remove the cell borders from the worksheet, use the following code to create a custom style which derives from the built-in style named STYLES.EXCEL but has the BORDERSTYLE attribute set to HIDDEN. Note the use of the ODS PATH statement to ensure that this custom style can be found when it is applied using the STYLE= option on the ODS EXCEL statement.

```
ods path(prepend) work.templat(update);
proc template;
  define style styles.mystyle;
    parent=styles.excel;
    class data, header, table, dataemphasis, linecontent /
      borderstyle=hidden;
  end;
run;
ods excel file="c:\example.xlsx" style=styles.mystyle;
     <output-generating code code goes here...>;
ods excel close;
```

As an alternative, to modify the appearance of the cell borders, change the values of the BORDERSTYLE, BORDERWIDTH, and/or BORDERCOLOR attributes as desired. The example below creates an Excel worksheet with solid red cell borders that are 3 pixels in thickness.

```
ods path(prepend) work.templat(update);
proc template;
  define style styles.mystyle;
    parent=styles.excel;
    class data, header, table, dataemphasis, linecontent /
      borderstyle=solid
      borderwidth=3px
      bordercolor=red;
  end;
run;
ods excel file="c:\example.xlsx" style=styles.mystyle;
     <output-generating code code goes here...>;
ods excel close;
```

CREATING EXCEL FORMULAS FROM SAS

Formulas are one of the most powerful features of Excel, and the ODS EXCEL destination provides a way to build Excel formulas directly from SAS. In general, ODS EXCEL will format any character data as a text string in Excel. However, if the FORMULAS sub-option of the OPTIONS option is set to "YES" then any string beginning with an equal sign will be interpreted as an Excel formula.

```
ods excel file="c:\example.xlsx" options(formulas="YES");
```

WORKING WITH DATES AND TIMES

SAS and Microsoft Excel use different methods to represent date and time values. SAS stores dates as the number of days since January 1, 1960, while Microsoft Excel instead counts the days since January 1, 1900.

Times in SAS are represented by the number of seconds since midnight, while Excel converts the time of day into a decimal value representing the fraction of a day that would have passed at that clock time. In other words, Excel stores noon as 0.5 (halfway through the day) and SAS stores noon as 43,200.

When creating an Excel spreadsheet using the ODS EXCEL destination that includes date and/or time values, it's important to make sure that these values are correctly converted. To do this, simply be sure that an appropriate date or time format is applied in SAS to any variables which are used in your Excel output. Unformatted values will not convert correctly since ODS EXCEL will be unaware what they represent.

CUSTOMIZING ODS EXCEL OUTPUT WITH STYLES

The previous section described how to create an Excel workbook directly from SAS using the ODS EXCEL destination and how to modify some basic properties and features of the resulting Excel file. ODS styles take this one step further by providing rich functionality for the complete customization of nearly every aspect of the appearance of your Excel workbook. This includes colors, fonts, borders, backgrounds, and much more. We will briefly discuss three ways to access this functionality: built-in styles, custom styles, and in-line styling.

USING BUILT-IN STYLES

The SAS system includes a library of built-in styles that are ready for use. Each of these styles represents a complete set of visual attributes that can be applied to any ODS output, whether that's an Excel spreadsheet or another ODS destination. Some of the built-in styles that are included with SAS are listed below:

- STYLES.BANKER
- STYLES.BRICK
- STYLES.HARVEST
- STYLES.OCEAN
- And many more…

These built-in styles are stored in the SASHELP.TMPLMST item store. To see a list of available built-in styles, submit the following PROC TEMPLATE code:

```
proc template;
   path sashelp.tmplmst;
   list styles;
run;
```

To apply one of these styles to Excel output generated through the ODS EXCEL destination, utilize the STYLE= option on the ODS EXCEL statement when opening the destination.

```
      ods excel file="c:\example.xlsx" style=styles.ocean;
```

CREATING YOUR OWN STYLES

In addition to using the wide variety of built-in styles described above, ODS also allows you to define your own custom styles using PROC TEMPLATE. This gives you the flexibility to specify the values of each of the different visual attributes that control every aspect of how your Excel output appears.

You can either create a new style completely from scratch, or you can start with an existing style and modify it as desired. The following PROC TEMPLATE code creates a new style called STYLES.MYSTYLE which inherits the properties of the built-in style STYLES.PEARL and changes the value of a single attribute.

```
      proc template;
           define style styles.mystyle;
                parent = styles.pearl;
                class data / color=red;
           end;
      run;
```

This style can then be applied to ODS EXCEL output as before:

```
      ods excel file="c:\example.xlsx" style=styles.mystyle;
```

PROC TEMPLATE is a complex procedure with an extensive array of available classes, elements, and attributes. Detailed information can be found in the PROC TEMPLATE documentation in *SAS 9.4 Output Delivery System: Procedures Guide, Third Edition* (2016), or refer to Smith (2013) for a thorough treatment of the topic.

USING IN-LINE STYLING

The final way to employ ODS styles is using in-line styling. In-line styling provides a convenient way to override the appearance of individual elements of your output without the need to use PROC TEMPLATE to create a complete style template. This is done by specifying one or more pairs of attributes and values using the STYLE= option, which is available on specific statements in various procedures including the PRINT, REPORT, and TABULATE procedures, among others.

Within PROC REPORT, in-line styling can be applied to several different report locations. These locations are named COLUMN, HEADER, SUMMARY, REPORT, LINES, and CALLDEF. Selecting one of these locations allows you to control which portions of the report output will be affected by the in-line style. If a location is not specified, the in-line style will be applied to a default location. The default location is different depending on which PROC REPORT statement contains the in-line style.

Statement	Default Location
PROC REPORT	REPORT
DEFINE	COLUMN and HEADER
BREAK / RBREAK	SUMMARY and LINES
COMPUTE	LINES

The following PROC REPORT code shows how in-line styling can be applied to modify various attributes in various report locations. The output is directed to an Excel spreadsheet using the ODS EXCEL destination, but the same PROC REPORT code could have been used with most other ODS destinations as well.

```
ods excel file="c:\example.xlsx";
proc report data=sashelp.cars
            style(report)=[cellpadding=3px];
   define make  / style(column)=[color=red];
   define model / style(header)=[color=green
                                 font_size=14pt];
   define type  / style=[background_color=blue
                         color=white];
run;
ods excel close;
```

Refer to Eslinger (2018) for a thorough explanation of customizing PROC REPORT output using ODS styles.

CONCLUSION

The ODS EXCEL destination provides a powerful and flexible way to create Microsoft Excel workbooks directly from SAS. The use of ODS styles and the various ODS EXCEL options provides for extensive customization of nearly every aspect of the resulting output. This integration between SAS and Microsoft Excel allows one to have the best both worlds – the power and functionality of the SAS system along with the flexibility and ubiquity of Microsoft Excel.

REFERENCES

Eslinger, Jane. 2018. *The SAS® Programmer's PROC REPORT Handbook: ODS Companion*. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2016. *SAS® 9.4 Output Delivery System: Procedures Guide, Third Edition*. Cary, NC: SAS Institute Inc. https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/odsproc/titlepage.htm

SAS Institute Inc. 2016. *SAS® 9.4 Output Delivery System: User's Guide, Fifth Edition*. Cary, NC: SAS Institute Inc. https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/odsug/titlepage.htm

Smith, Kevin D. 2013. *PROC TEMPLATE Made Easy: A Guide for SAS® Users*. Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

Joshua M. Horstman
Nested Loop Consulting
317-721-1009
josh@nestedloopconsulting.com