Paper SAS4243-2020

Creating Custom Microsoft Excel Workbooks Using the SAS® Output Delivery System, Part 1

Vincent DelGobbo, SAS Institute Inc.

ABSTRACT

This paper explains how to use Base SAS® software to create custom multi-sheet Microsoft Excel workbooks. You learn step-by-step techniques for quickly and easily creating attractive multi-sheet Excel workbooks that contain your SAS® output by using the SAS® Output Delivery System (ODS) Report Writing Interface (RWI) and the ODS destination for Excel. The techniques can be used regardless of the platform on which SAS software is installed. You can even use them on a mainframe! Creating and delivering your workbooks on demand and in real time using SAS server technology is discussed. Although the title is similar to previous presentations by this author, this presentation contains new and revised material not previously presented.

INTRODUCTION

This paper explains how to use the ODS destination for Excel, the ODS Report Writing Interface (RWI), and the REPORT procedure to create the Excel workbook shown in Figure 1 and Figure 2.

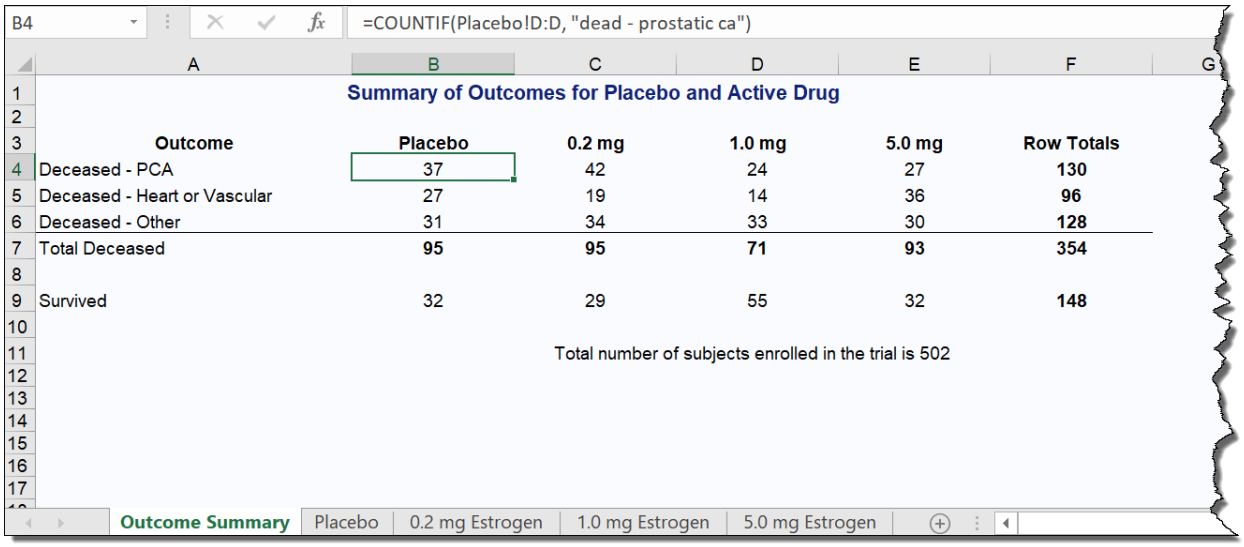
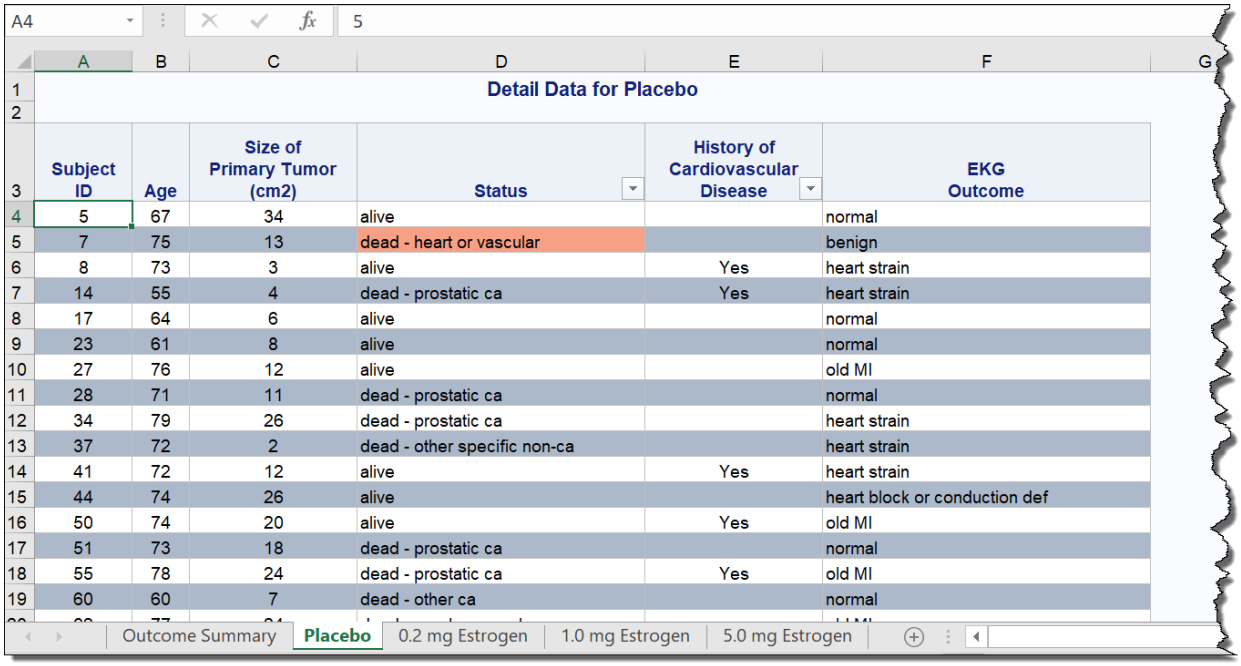


Figure 1. Worksheet Generated Using the ODS Report Writing Interface



Detail Data for Placebo

Subject ID	Age	Size of Primary Tumor (cm2)	Status	History of Cardiovascular Disease	EKG Outcome
5	67	34	alive		normal
7	75	13	dead - heart or vascular		benign
8	73	3	alive	Yes	heart strain
14	55	4	dead - prostatic ca	Yes	heart strain
17	64	6	alive		normal
23	61	8	alive		normal
27	76	12	alive		old MI
28	71	11	dead - prostatic ca		normal
34	79	26	dead - prostatic ca		heart strain
37	72	2	dead - other specific non-ca		heart strain
41	72	12	alive	Yes	heart strain
44	74	26	alive		heart block or conduction def
50	74	20	alive	Yes	old MI
51	73	18	dead - prostatic ca		normal
55	78	24	dead - prostatic ca	Yes	old MI
60	60	7	dead - other ca		normal

Figure 2. Worksheet Generated Using the REPORT Procedure

The REPORT procedure creates four worksheets (sample shown in Figure 2) containing detailed clinical trial data comparing four treatments for patients with prostate cancer: a placebo, and three different doses (0.2 mg, 1.0 mg, and 5.0 mg) of estrogen (Andrews and Herzberg 1985). Different background colors are applied to alternating rows to make them easier to read, and patients who died due to cardiovascular disease, a possible side effect of the treatment, are highlighted in orange (Byar and Green 1980; Bailar and Byar 1970). Worksheet names are automatically created based on the treatment, and Excel AutoFilters are included for some columns to assist in filtering the data.

The worksheet summarizing the outcomes (Figure 1) was created using the ODS Report Writing Interface. Each data cell contains an Excel formula that calculates the cell value based on information from a detailed data worksheet. For example, the value in cell B4 is calculated from data in the "Placebo" worksheet using this formula:

```
=COUNTIF(Placebo!D:D, "dead - prostatic ca")
```

The formulas automatically recalculate the values when changes are made to the "Status" column of the detailed data worksheets.

All formatting and layout are performed by SAS; there is no need to "hand-edit" the Excel workbook. You simply use Excel to open the file created by SAS.

The code in this paper was tested using SAS® 9.4M6 and Microsoft Excel 2016 software. You can download the sample data and code here:

https://github.com/sascommunities/sas-global-forum-2020/tree/master/papers/4243-2020-Delgobbo

Download the ZIP file and then view the information in the "ReadMe.txt file".

SAMPLE DATA

Table 1 presents abbreviated information about the PROSTATECANCER SAS data set used to create the Excel workbook shown in Figure 1 and Figure 2. An asterisk (*) is used as a split character in some variable labels to control text wrapping in the column headings.

Variable Name	Variable Label	Variable Type	Typical Values
RX	Drug	Numeric	1, 2, 3, or 4
PATNO	Subject*ID	Numeric	1 - 506
AGE	Age in*Years	Numeric	48 - 89
SZ	Size*of*Primary Tumor*(cm2)	Numeric	0 - 69
STATUS	Status	Character	alive, dead - prostatic ca
HX	History of*Cardiovascular*Disease	Numeric	0 or 1
EKG	EKG Outcome	Character	normal, heart strain

Table 1. Representative Data Values in the PROSTATECANCER SAS Data Set

The "rx" format is used with the RX variable, and the "boolean" format is used with the HX variable:

```
proc format;
  value rx 1 = 'Placebo'
           2 = '0.2 mg Estrogen'
           3 = '1.0 mg Estrogen'
           4 = '5.0 mg Estrogen';

  value boolean 0 = ' '
                1 = 'Yes';
run; quit;
```

When the REPORT procedure is run against the PROSTATECANCER data set, worksheet names corresponding to the formatted values of the BY variable RX are automatically created. For example, "Placebo" and "1.0 mg Estrogen" (Figure 2).

OUTPUT DELIVERY SYSTEM (ODS) BASICS

ODS is the part of Base SAS software that enables you to generate different types of output from your procedure and DATA step code. An ODS *destination* controls the type of output that is generated (HTML, RTF, PDF, and so on). An ODS *style* controls the appearance of the output.

The Excel workbook shown in Figure 1 and Figure 2 was created using the ODS EXCEL destination and the HTMLBLUE ODS style supplied by SAS. Here are the general statements to generate an Excel XLSX file:

```
❶ ods _all_ close;

❷ ods excel file='directory-location\file-name.xlsx' style=style-name;
    * Your SAS code here;
❸ ods excel close;
```

The first ODS statement (❶) closes all destinations that are open because we want to generate only Excel XLSX output.

The second ODS statement (❷) uses the EXCEL destination to generate the output and then store it in a file (SAS Institute Inc. 2020e). The STYLE option controls the appearance of the output, such as the font and color scheme. To see a list of ODS styles that are available for use at your site, submit the following SAS code.

```
ods _all_ close;
ods listing;
proc template; list styles; run; quit;
```

To find the SAS code that generates sample output for the ODS styles available on your system, select the Full Code tab in SAS Sample 36900 (SAS Institute Inc. 2009).

The third ODS statement (❸) closes the EXCEL destination and releases the file so that it can be opened with Microsoft Excel.

UNDERSTANDING AND USING ODS STYLE OVERRIDES

You can alter the appearance of specific parts of your PRINT, REPORT, and TABULATE procedure output by using style overrides. These specific parts of your SAS output are called *locations*. Figure 3 shows the locations of the REPORT procedure output (SAS Institute Inc. 2019e).

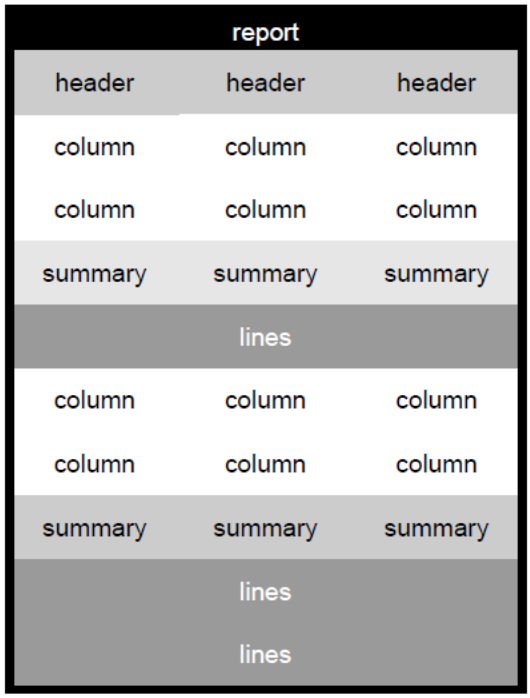


Figure 3. Style Locations for the REPORT Procedure

Here is the most common format for specifying style overrides:

```
style(location)=[attribute-name1=value1
                 attribute-name2=value2 ...]
```

The COLUMN location applies to the data cells and is the location that we use with PROC REPORT.

You can use a style override in a PROC statement to change the appearance of *all* columns in your output:

```
proc report style(column)=[background=yellow font_size=10pt just=left] ...
```

The code specifies that all data cells in the report have a yellow background, and use left-justified, 10-point text.

To change the appearance of data cells for individual variables in your report, specify the style override in a DEFINE statement:

```
define myvar / style(column)=[just=center] ... ;
```

The CALL DEFINE statement in PROC REPORT can also be used to apply a style override to the data cells. The general syntax is:

```
call define(column-id, 'style', 'style=[attribute-name=value ...]')
```

Refer to the Creating the Detailed Data Worksheets section to see the code that applies style overrides to our output. The ODS documentation provides a full listing of style attributes (SAS Institute Inc. 2020h).

UNDERSTANDING AND USING THE EXCEL DESTINATION OPTIONS

The EXCEL destination supports options that control both the appearance and functionality of the workbook. Many of these options are simply tied directly into existing Excel options or features. For example, the SHEET_NAME option specifies the worksheet name.

Options 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 an option remains in effect until the EXCEL destination is closed or the option is set to another value. Because multiple ODS statements are allowed, it is good practice, in terms of functionality and code readability, to explicitly reset an option to its default value when you are finished using it.

Here is an example:

```
ods excel file='directory-location\file-name.xlsx' style=style-name ... ;

ods excel options(option-name='some-value');
  * Some SAS code here;
ods excel options(option-name='default-value');
  * Other SAS code here;
ods excel close;
```

When specifying multiple ODS statements as shown above, specify the FILE and STYLE options only in the initial ODS statement.

SETTING UP THE PROGRAM ENVIRONMENT

The code below closes all ODS destinations and creates formats used with some of the variables in the sample data.

```
*  Close all ODS destinations, and then open when needed;

ods _all_ close;

*  Library for input data;

libname sample 'directory-location' access=read;

*;
*  Create formats to make drug codes
*  and Boolean values more user-friendly
*;

proc format;
  value rx 1 = 'Placebo'
           2 = '0.2 mg Estrogen'
           3 = '1.0 mg Estrogen'
           4 = '5.0 mg Estrogen';

  value boolean 0 = ' '
                1 = 'Yes';
run; quit;
```

CREATING THE DETAILED DATA WORKSHEETS

This code creates the detailed data worksheets shown in Figure 2.

```
❶ ods excel file='directory-location\ProstateCancer.xlsx' style=HTMLBlue
    options(embedded_titles='yes'
            suppress_bylines='yes');

❷ title 'Detail Data for #byval(rx)';
  footnote;

❸ ods excel options(sheet_interval='bygroup'
                    sheet_name='#byval(rx)'
                    autofilter='4-5');

  * One worksheet created for each distinct BY value;

❹ proc report data=sample.ProstateCancer nowindows split='*'
     style(column)=[just=center];
    by rx;
    column PatNo Age SZ Status HX EKG;

❺   define PatNo  / display 'Subject*ID';
    define Age    / display 'Age';
    define SZ     / display 'Size of*Primary Tumor*(cm2)';
    define Status / display style(column)=[just=left];
    define HX     / display 'History of*Cardiovascular*Disease'
                      format=boolean.;
    define EKG    / display 'EKG*Outcome' style(column)=[just=left];

❻   compute PatNo;
      *  Change background color for alternating rows;
      RowNum+1;
      if (mod(RowNum, 2) eq 0)
      then call define(_row_, 'style', 'style=[background=#acb9ca]');
    endcomp;
```

```
❼   compute Status;
      *  Change background color for certain cells;
      if (Status eq 'dead - heart or vascular')
      then call define('Status', 'style', 'style=[background=#f7a085]');
    endcomp;

❽   format rx rx.;
  run; quit;

  ods excel close;
```

The EXCEL destination generates the ProstateCancer.xlsx file and the HTMLBLUE style controls the appearance of the output (❶). Options specified in this ODS statement apply to all worksheets because their values are not changed later in the code. #BYVAL (❷) substitutes the current value of the REPORT procedure BY variable RX into the TITLE statement (SAS Institute Inc. 2019a). Here is an example:

```
title 'Detail Data for 0.2 mg Estrogen';
```

We use #BYVAL with the SHEET_NAME option (❸) to automatically name the worksheets based on the BY value of the RX variable.

Table 2 briefly explains the Excel-specific options used in our code (❶ and ❸).

Option	Description
SHEET_INTERVAL	A new worksheet is created for each BY group.
SHEET_NAME	The value of the BY variable RX is used for worksheet names.
EMBEDDED_TITLES	Title text appears in the workbook, instead of the print header.
SUPPRESS_BYLINES	BY line text is not included in the output.
AUTOFILTER	Excel AutoFilters are applied to columns 4 (STATUS) and 5 (HX).

Table 2. Excel Destination Options Used to Create the Worksheet in Figure 2

Refer to the ODS documentation for detailed information about these and other options (SAS Institute Inc. 2020e).

PROC REPORT (❹) is run with a BY statement and creates four worksheets, one for each distinct value of the variable RX. An ODS style override specifies that the data cell text for all variables should be centered. Style overrides in the DEFINE statements (❺) specify that the data cell text for the STATUS and EKG variables should be left-justified.

Adding a background color to alternating rows makes the worksheets more attractive and easier to read. The code in the first COMPUTE block (❻) applies a style override that changes the background color of even numbered data rows (SAS Institute Inc. 2020d).

The second block (❼) applies a style override to specific cells based on a rule. This general technique is referred to as "traffic lighting" and is used to emphasize data that meet the rule criteria. In our case, we want to draw attention to patients who died due to cardiovascular disease by changing the cell background color to orange.

The user-defined formats, "boolean" and "rx", are applied to the HX and RX variables, respectively (❺ and ❽).

This author's earlier paper provides additional information about this PROC REPORT code (DelGobbo 2011).

INTRODUCTION TO THE REPORT WRITING INTERFACE (RWI)

The ODS Report Writing Interface allows you to use DATA step programming to create highly customized ODS output. You have nearly complete control of text placement and formatting, and can create output that cannot be created by the PRINT and REPORT procedures.

The RWI uses object-oriented programming (OOP) techniques. You do not need to be an expert in OOP, but you should be familiar with the following terminology (SAS Institute Inc. 2020i).

class
: is a template for an object. A class includes data that describes the object's characteristics (such as attributes or instance variables), as well as the operations (methods) that the object can perform.

instance
: is the run-time initialization of the class object attributes and methods.

object
: is any entity that can be manipulated by the commands of a programming language. Examples are values, variables, functions, and data structures.

method
: in object-oriented methodology, is an operation that is defined for a class and can be executed by an object that is created from that class.

Methods are like SAS CALL routines, except they are executed using "dot notation":

```
object.method(arguments)
```

In the DATA step code below, odsout is the class that you want to instantiate. The DECLARE statement creates an instance of the ODSOUT class and creates the ODSobj object. You can specify any valid SAS name for the object. The FORMAT_TEXT method is executed with the DATA argument.

```
ods _all_ close;

ods html5 file='directory-location\RWI.htm';
ods pdf   file='directory-location\RWI.pdf';
ods excel file='directory-location\RWI.xlsx';

data _null_;
declare odsout ODSobj();
ODSobj.format_text(data: 'My first RWI program!');
run;

ods _all_ close;
```

The text "My first RWI program!" is written to the HTML, PDF, and XLSX files specified in the ODS statements.

The methods that we use in our code to create the tabular output of Figure 1 are listed in Table 3.

Method	Description
TABLE_START	Specifies the start of a table.
ROW_START	Specifies the beginning of a row.
FORMAT_CELL	Adds content to and formats a cell.
ROW_END	Specifies the end of a row. Required if ROW_START was used.
TABLE_END	Specifies the end of a table. Required if TABLE_START was used.

Table 3. ODS Output Object Methods Used to Create the Worksheet of Figure 1

The FORMAT_CELL method writes data to cells and controls their appearance (SAS Institute Inc. 2020c). We use these four arguments in our code:

DATA	Specifies the data to display. This argument is required.
JUST	Specifies the horizontal text justification. Values are L, C, or R.
STYLE_ATTR	Specifies the style attributes to override.
COLUMN_SPAN	Specifies the number of columns that the cell occupies.

This code creates a table with three rows and two columns, and illustrates the usage of the methods:

```
ods _all_ close;

ods excel file='directory-location\RWI.xlsx';

data _null_;
declare odsout ODSobj();

ODSobj.table_start();

  ODSobj.row_start(); *  Row 1;
    *  Cells A1 - A2;
    ODSobj.format_cell(data: 'Column Headings',
                       style_attr: 'fontweight=bold',
                       column_span: 2);
  ODSobj.row_end();

  ODSobj.row_start(); *  Row 2;
    *  Cell A2;
    ODSobj.format_cell(data: 'Heading 1',
                       style_attr: 'fontweight=bold');
    *  Cell B2;
    ODSobj.format_cell(data: 'Heading 2',
                       style_attr: 'fontweight=bold');
  ODSobj.row_end();

  ODSobj.row_start(); *  Row 3;
    *  Cell A3;
    ODSobj.format_cell(data: 'Data 1',
                       just: 'L');
    *  Cell B3;
    ODSobj.format_cell(data: 'Data 2',
                       just: 'L');
  ODSobj.row_end();
```

```
ODSobj.table_end();

run;

ods excel close;
```

We use the TABLE_START method to begin the table, and the ROW_START and ROW_END methods to create rows. The FORMAT_CELL method, used in the first row, creates one cell that spans two columns. Remaining FORMAT_CELL method calls create non-spanning cells. Text justification and font weight are specified using the JUST and STYLE_ATTR arguments, respectively. We end the table by calling the TABLE_END method.

Figure 4 displays the results.

	A	B
1	**Column Headings**	
2	**Heading 1**	**Heading 2**
3	Data 1	Data 2

Figure 4. Worksheet in the RWI.xlsx File

The ODS documentation provides a full listing of style attributes that can be specified in the STYLE_ATTR argument (SAS Institute Inc. 2020c) and detailed information about the methods (SAS Institute Inc. 2020g).

CREATING THE OUTCOME SUMMARY WORKSHEET

This section provides step-by-step instructions for creating the "Outcome Summary" worksheet of Figure 1:

1. Design the worksheet.
2. Run code to create the worksheet with hard-coded data.
3. Replace some hard-coded data with summation formulas.
4. Apply style overrides to change the appearance.
5. Replace remaining hard-coded data with Excel formulas.

DESIGNING THE WORKSHEET

The worksheet that we need to create is shown in Figure 5:

	A	B	C	D	E	F
1	**Summary of Outcomes for Placebo and Active Drug**					
2						
3	**Outcome**	**Placebo**	**0.2 mg**	**1.0 mg**	**5.0 mg**	**Row Totals**
4	Deceased - PCA	37	42	24	27	**130**
5	Deceased - Heart or Vascular	27	19	14	36	**96**
6	Deceased - Other	31	34	33	30	**128**
7	Total Deceased	**95**	**95**	**71**	**93**	**354**
8						
9	Survived	32	29	55	32	**148**
10						
11			Total number of subjects enrolled in the trial is 502			

Figure 5. Worksheet to Create Using the RWI

A TITLE statement displays informative text at the top of the worksheet and the RWI is used to create the remaining content.

Because Excel is cell-based, it is helpful to superimpose a grid onto your desired output to better visualize the table structure (Figure 6).

	A	B	C	D	E	F
1	**Summary of Outcomes for Placebo and Active Drug**					
2						
3	**Outcome**	**Placebo**	**0.2 mg**	**1.0 mg**	**5.0 mg**	**Row Totals**
4	Deceased - PCA	37	42	24	27	**130**
5	Deceased - Heart or Vascular	27	19	14	36	**96**
6	Deceased - Other	31	34	33	30	**128**
7	Total Deceased	**95**	**95**	**71**	**93**	**354**
8						
9	Survived	32	29	55	32	**148**
10						
11		Total number of subjects enrolled in the trial is 502				

Figure 6. Worksheet to Create Using the RWI With Superimposed Grid

The RWI code must create a table with these features of Figure 6:

- Nine rows and six columns.
- The sixth and eight rows must be blank, and the cells must span six columns.
- All text must be centered, except the text in the "Outcome" column.
- Column headings and summation values have bold text.
- A solid line appears before the summation values for deceased subjects.
- The second cell in the last row must span five columns.
- Excel formulas calculate the values in the data cells.

CODE TO CREATE THE WORKSHEET WITH HARD-CODED VALUES

This code creates the worksheet with the correct structure. The code to create the detail data worksheets, discussed earlier, follows the RWI code.

```
ods excel file='directory-location\ProstateCancer.xlsx' style=HTMLBlue
  options(embedded_titles='yes'
          suppress_bylines='yes');

title 'Summary of Outcomes for Placebo and Active Drug';
footnote;

ods excel options(sheet_name='Outcome Summary');

data _null_;

*  Declare the ODS output object;

declare odsout ODSobj();

*  Start the table;

ODSobj.table_start();
```

```
*  Column heading row;

ODSobj.row_start();  *  Cells A3 - F3;
  ODSobj.format_cell(data: 'Outcome');

  ODSobj.format_cell(data: 'Placebo');

  ODSobj.format_cell(data: '0.2 mg');

  ODSobj.format_cell(data: '1.0 mg');

  ODSobj.format_cell(data: '5.0 mg');

  ODSobj.format_cell(data: 'Row Totals');
ODSobj.row_end();

*  Data row 1;

ODSobj.row_start();  *  Cells A4 - F4;
  ODSobj.format_cell(data: 'Deceased - PCA',
                     just: 'L');

  ODSobj.format_cell(data: 37);

  ODSobj.format_cell(data: 42);

  ODSobj.format_cell(data: 24);

  ODSobj.format_cell(data: 27);

  ODSobj.format_cell(data: 130);
ODSobj.row_end();

*  Data row 2;

ODSobj.row_start();  *  Cells A5 - F5;
  ODSobj.format_cell(data: 'Deceased - Heart or Vascular',
                     just: 'L');

  ODSobj.format_cell(data: 27);

  ODSobj.format_cell(data: 19);

  ODSobj.format_cell(data: 14);

  ODSobj.format_cell(data: 36);

  ODSobj.format_cell(data: 96);
ODSobj.row_end();

*  Data row 3;

ODSobj.row_start();  *  Cells A6 - F6;
  ODSobj.format_cell(data: 'Deceased - Other',
                     just: 'L');

  ODSobj.format_cell(data: 31);

  ODSobj.format_cell(data: 34);
```

```
  ODSobj.format_cell(data: 33);

  ODSobj.format_cell(data: 30);

  ODSobj.format_cell(data: 128);
ODSobj.row_end();

*  Total Deceased row;

ODSobj.row_start();  *  Cells A7 - F7;
  ODSobj.format_cell(data: 'Total Deceased',
                     just: 'L');

  ODSobj.format_cell(data: 95);

  ODSobj.format_cell(data: 95);

  ODSobj.format_cell(data: 71);

  ODSobj.format_cell(data: 93);

  ODSobj.format_cell(data: 354);
ODSobj.row_end();

*  Blank row;

ODSobj.row_start();  *  Merged Cells A8 - F8;
  ODSobj.format_cell(data: '',
                     column_span: 6);
ODSobj.row_end();

*  Data row 4;

ODSobj.row_start();  *  Cells A9 - F9;
  ODSobj.format_cell(data: 'Survived',
                     just: 'L');

  ODSobj.format_cell(data: 32);

  ODSobj.format_cell(data: 29);

  ODSobj.format_cell(data: 55);

  ODSobj.format_cell(data: 32);

  ODSobj.format_cell(data: 148);
ODSobj.row_end();

*  Blank row;

ODSobj.row_start();  *  Merged Cells A10 - F10;
  ODSobj.format_cell(data: '',
                     column_span: 6);
ODSobj.row_end();

*  Footer row;

ODSobj.row_start();  *  Cell A11, and Merged Cells B11 - F11;
  ODSobj.format_cell(data: '');
```

```
  ODSobj.format_cell(data: 'Total number of subjects enrolled in the trial
is 502',
                     column_span: 5);
ODSobj.row_end();

*  End the table;

ODSobj.table_end();

run;

*  Code to create the detail data worksheets follows;

title 'Detail Data for #byval(rx)';
footnote;

ods excel options(sheet_interval='bygroup'
                  sheet_name='#byval(rx)'
                  autofilter='4-5');

* One worksheet created for each distinct BY value;

proc report data=sample.ProstateCancer ... ;
  ... ;
run; quit;

ods excel close;
```

All text is hard-coded using the DATA argument, and the JUST argument left-justifies some text. The results are shown in Figure 7.

	A	B	C	D	E	F
1	**Summary of Outcomes for Placebo and Active Drug**					
2						
3	Outcome	Placebo	0.2 mg	1.0 mg	5.0 mg	Row Totals
4	Deceased - PCA	37	42	24	27	130
5	Deceased - Heart or Vascular	27	19	14	36	96
6	Deceased - Other	31	34	33	30	128
7	Total Deceased	95	95	71	93	354
8						
9	Survived	32	29	55	32	148
10						
11		Total number of subjects enrolled in the trial is 502				

Figure 7. Outcome Summary Worksheet with Hard-Coded Values

The output resembles Figure 1 with these differences:

- Hard-coded values, not Excel formulas, are used in the data cells.
- Column headings and summation values do not have bold text.
- There is no solid line before the summation values for deceased subjects.
- The table has a border lines and a white background.

These issues are corrected in the sections that follow.

ADDING SUMMATION FORMULAS

Excel formulas begin with an equal sign ("="), and usually contain references to other cells in a worksheet. An advantage of using formulas instead of hard-coded values is that the values are recalculated when the cells referenced in the formulas change.

We want to use Excel formulas to compute the values in the "Row Totals" column and in the "Total Deceased" row. The existing hard-coded values must be replaced with the appropriate formulas. The formulas used in the "Row Totals" column are shown in Table 4.

Cell Reference	Hard-Coded Value	Excel Formula
F4	130	=SUM(B4:E4)
F5	96	=SUM(B5:E5)
F6	128	=SUM(B6:E6)
F7	354	=SUM(B7:E7)
F9	148	=SUM(B9:E9)

Table 4. Excel Formulas for the "Row Totals" Column

Here is an example of replacing a hard-coded value with a formula for the first cell (F4) in the "Row Totals" column:

```
*  Data row 1;

ODSobj.row_start();  *  Cells A4 - F4;
  ODSobj.format_cell(data: 'Deceased - PCA',
                     just: 'L');          * Cell A4;

  ODSobj.format_cell(data: 37);           * Cell B4;

  ODSobj.format_cell(data: 42);           * Cell C4;

  ODSobj.format_cell(data: 24);           * Cell D4;

  ODSobj.format_cell(data: 27);           * Cell E4;

  ODSobj.format_cell(data: =SUM(B4:E4));  * Cell F4;
ODSobj.row_end();
```

Similar changes need to be made to the remaining cells listed in Table 4.

The formulas used in the "Total Deceased" row are shown in Table 5.

Cell Reference	Hard-Coded Value	Excel Formula
B7	95	=SUM(B4:B6)
C7	95	=SUM(C4:C6)
D7	71	=SUM(D4:D6)
E7	93	=SUM(E4:E6)

Table 5. Excel Formulas for the "Total Deceased" Row

Here is the complete code to replace hard-coded values with formulas for cells B7 - E7. The formula shown for cell F7 was specified earlier when adding the formulas for the "Row Totals" column.

```
*  Total Deceased row;

ODSobj.row_start();  *  Cells A7 - F7;
  ODSobj.format_cell(data: 'Total Deceased',
                     just: 'L');            * Cell A7;

  ODSobj.format_cell(data: '=SUM(B4:B6)');  * Cell B7;

  ODSobj.format_cell(data: '=SUM(C4:C6)');  * Cell C7;

  ODSobj.format_cell(data: '=SUM(D4:D6)');  * Cell D7;

  ODSobj.format_cell(data: '=SUM(E4:E6)');  * Cell E7;

  ODSobj.format_cell(data: '=SUM(B7:E7)');  * Cell F7;
ODSobj.row_end();
```

APPLYING STYLE OVERRIDES TO CHANGE THE APPEARANCE

In this section we use style overrides with the FORMAT_CELL method to make some of the text bold, and change the background and border cell colors of the table.

The STYLE_ATTR argument is used to specify style overrides for a cell. The general syntax is:

```
style_attr: <'attribute-name=attribute-value' | character-variable>
```

Code samples ❶ and ❷ are equivalent, and result in the word "Outcome" having bold text:

```
❶ ODSobj.format_cell(data: 'Outcome',
                     style_attr: 'fontweight=bold');


❷ bold_text = 'fontweight=bold';

  ODSobj.format_cell(data: 'Outcome',
                     style_attr: bold_text);
```

We use syntax ❷ because the code is easier to read when more than one attribute is specified. When specifying multiple attributes in this way, use the CATX function to construct a character string with the name/value pairs separated by a blank space (SAS Institute Inc. 2020b):

```
bold_text        = 'fontweight=bold';
border_color     = 'bordercolor=#fafbfe';
background_color = 'backgroundcolor=#fafbfe';

ODSobj.format_cell(data: 'Outcome',
                   style_attr: catx(' ', bold_text,
                                         border_color,
                                         background_color) );
```

Making Some of the Text Bold

The initial code that creates the column headings must be changed because the values should have bold text. We define the BOLD_TEXT variable and then use it with the STYLE_ATTR argument to specify that the column headings have bold text:

```
bold_text = 'fontweight=bold';

*  Column heading row;

ODSobj.row_start();  *  Cells A3 - F3;
  ODSobj.format_cell(data: 'Outcome',
                     style_attr: bold_text);

  ODSobj.format_cell(data: 'Placebo',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '0.1 mg',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '1.0 mg',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '5.0 mg',
                     style_attr: bold_text);

  ODSobj.format_cell(data: 'Row Totals',
                     style_attr: bold_text);
ODSobj.row_end();
```

Similar changes are needed, so the summation values have bold text. Here is the code for the "Total Deceased" row:

```
bold_text  = 'fontweight=bold';

*  Total Deceased row;

ODSobj.row_start();
  ODSobj.format_cell(data: 'Total Deceased',
                     just: 'L');

  ODSobj.format_cell(data: '=SUM(B4:B6)',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '=SUM(C4:C6)',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '=SUM(D4:D6)',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '=SUM(E4:E6)',
                     style_attr: bold_text);

  ODSobj.format_cell(data: '=SUM(B7:E7)',
                     style_attr: bold_text);
ODSobj.row_end();
```

The same STYLE_ATTR specification needs to be applied to the "Row Totals" data cells (F4 - F7, and F9).

Changing the Border Line and Background Colors

A solid black line is used to indicate that the values in the "Total Deceased" row are summation values (Figure 1).

The code for the "Total Deceased" row with this addition is shown below. We use the CATX function for cells that require more than one style attribute.

```
bold_text  = 'fontweight=bold';
solid_line = 'bordertopcolor=black';

*  Total Deceased row;

ODSobj.row_start();  *  Cells A7 - F7;
  ODSobj.format_cell(data: 'Total Deceased',
                     just: 'L',
                     style_attr: solid_line);

  ODSobj.format_cell(data: '=SUM(B4:B6)',
                     style_attr: catx(' ', bold_text, solid_line) );

  ODSobj.format_cell(data: '=SUM(C4:C6)',
                     style_attr: catx(' ', bold_text, solid_line) );

  ODSobj.format_cell(data: '=SUM(D4:D6)',
                     style_attr: catx(' ', bold_text, solid_line) );

  ODSobj.format_cell(data: '=SUM(E4:E6)',
                     style_attr: catx(' ', bold_text, solid_line) );

  ODSobj.format_cell(data: '=SUM(B7:E7)',
                     style_attr: catx(' ', bold_text, solid_line) );
ODSobj.row_end();
```

The worksheet with the style overrides applied up to this point is shown in Figure 8.

	A	B	C	D	E	F
1	**Summary of Outcomes for Placebo and Active Drug**					
2						
3	**Outcome**	**Placebo**	**0.2 mg**	**1.0 mg**	**5.0 mg**	**Row Totals**
4	Deceased - PCA	37	42	24	27	**130**
5	Deceased - Heart or Vascular	27	19	14	36	**96**
6	Deceased - Other	31	34	33	30	**128**
7	Total Deceased	**95**	**95**	**71**	**93**	**354**
8						
9	Survived	32	29	55	32	**148**
10						
11		Total number of subjects enrolled in the trial is 502				

Figure 8. The Outcome Summary Worksheet with Bold Text and Solid Line

The table cells have a white background and gray border lines. Specifying background and border line colors that match the document body background provides the desired appearance. Follow these steps in Excel to determine the document body background color.

1. Select the cell containing the title text.
2. In the Excel ribbon, select Home.
3. Select the drop-down arrow next to the "Fill Color" icon in the Font group, and then select More Colors from the Theme Colors dialog box.

4. Note the decimal values of the red, green, and blue color components in the Custom tab in the Colors dialog box.

The Excel dialog boxes are shown in Figure 9:

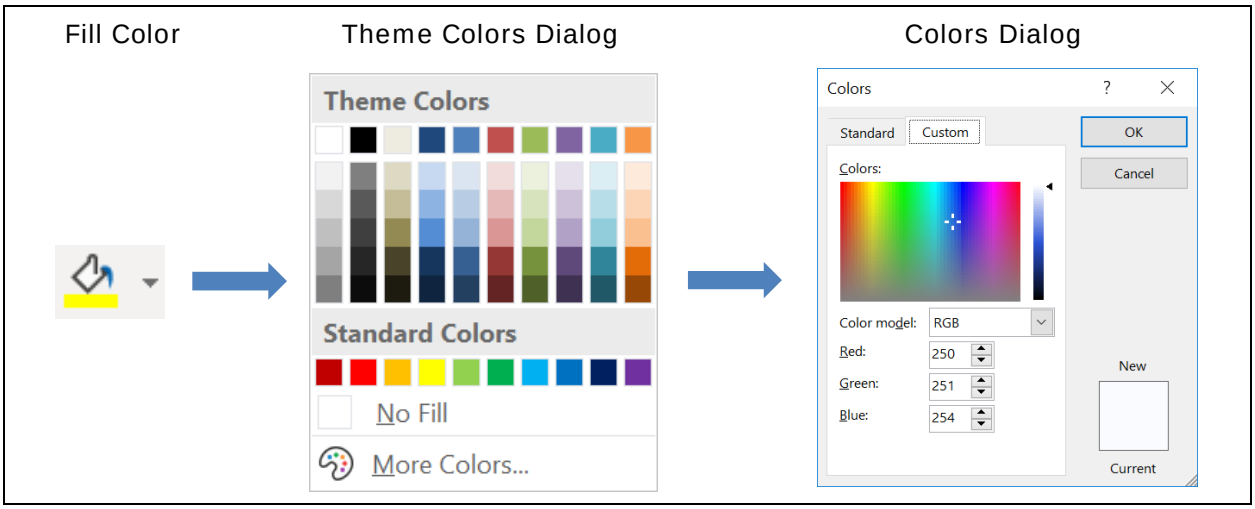


Figure 9. Steps to Determine the Document Body Background Color

ODS supports specifying only hexadecimal values for the red, green, and blue color components. This code uses the CAT function (SAS Institute Inc. 2020a) and HEX format (SAS Institute Inc. 2019b) to construct the color supported by ODS. Use the value displayed in the log in your style override code.

```
data _null_;
r = 250;
g = 251;
b = 254;
rgbcolor = cat('#', put(r, hex2.), put(g, hex2.), put(b, hex2.));
putlog 'NOTE: ' rgbcolor=;
run;

NOTE: rgbcolor=#FAFBFE
```

Define the BORDER_COLOR and BACKGROUND_COLOR variables, specifying "#fafbfe" for the values as shown below. Then specify the BORDER_COLOR and BACKGROUND_COLOR variables in the STYLE_ATTR argument for *all* cells. The code for the "Total Deceased" row is shown as an example of the needed changes:

```
*  Define variables for ODS style attributes;

bold_text        = 'fontweight=bold';
solid_line       = 'bordertopcolor=black';
border_color     = 'bordercolor=#fafbfe';
background_color = 'backgroundcolor=#fafbfe';

*  Total Deceased row;

ODSobj.row_start();  *  Cells A7 - F7;
  ODSobj.format_cell(data: 'Total Deceased',
                     just: 'L',
                     style_attr: catx(' ', solid_line, background_color,
                                      border_color) );
```

```
    ODSobj.format_cell(data: '=SUM(B4:B6)',
                       style_attr: catx(' ', bold_text, solid_line,
                                        background_color, border_color) );

    ODSobj.format_cell(data: '=SUM(C4:C6)',
                       style_attr: catx(' ', bold_text, solid_line,
                                        background_color, border_color) );

    ODSobj.format_cell(data: '=SUM(D4:D6)',
                       style_attr: catx(' ', bold_text, solid_line,
                                        background_color, border_color) );

    ODSobj.format_cell(data: '=SUM(E4:E6)',
                       style_attr: catx(' ', bold_text, solid_line,
                                        background_color, border_color) );

    ODSobj.format_cell(data: '=SUM(B7:E7)',
                       style_attr: catx(' ', bold_text, solid_line,
                                        background_color, border_color) );
  ODSobj.row_end();
```

The worksheet now looks like Figure 1, except there are still some hard-coded values. In the next section we replace the hard-coded values with formulas.

ADDING REMAINING FORMULAS

You cannot specify long formulas in the DATA argument in SAS 9.4M6 and earlier because they are interpreted as text. This problem has been corrected in SAS 9.4M7. If you encounter this problem, then you must use a workaround by specifying a style override using the TAGATTR attribute. Here is the general format for using TAGATTR to specify an Excel formula:

```
tagattr='formula:Excel-formula'
```

Be sure to use quotation marks around the value when using this attribute.

Footer Cell Formula

The Excel SUM function is used to compute the number of subjects enrolled in the trial by adding together the number of subjects who died and subjects who survived. Refer to Figure 1:

```
=SUM(F7,F9)
```

It is combined with the Excel CONCATENATE function to include text with the sum:

```
=CONCATENATE("Total number of subjects enrolled in the trial is ",
             SUM(F7,F9))
```

We specify a null value for the DATA argument in the FORMAT_CELL method, and then specify the formula using a style override:

```
formula = '=CONCATENATE("Total number of subjects enrolled in the trial is ",
           SUM(F7,F9))';

formula_attr = 'tagattr=' || quote('formula:' || formula);
```

```
ODSobj.format_cell(data: '',
                   column_span: 5,
                   style_attr: catx(' ', background_color, border_color,
                                    formula_attr) );
```

Excel requires that you use double quotation marks around the text in the formula. The SAS QUOTE function (SAS Institute Inc. 2020f) ensures that the TAGATTR attribute value is properly quoted.

Subject Count Formulas

The Excel COUNTIF function counts the number of cells in a range that meet a criterion. The general syntax is:

```
=COUNTIF(range-to-search, criterion)
```

We use the COUNTIF function to determine the number of subjects who survived and subjects who died in the four treatment groups. For example, this formula counts the number of patients in the "Placebo" group that died of prostate cancer (cell B4 in Figure 1):

```
=COUNTIF('Placebo'!D:D, "dead - prostatic ca")
```

The COUNTIF function searches all cells in column D of the "Placebo" worksheet (Figure 2) and returns the number of cells that contain the text "dead - prostatic ca". You must use single quotation marks when specifying the worksheet name.

The code for cell B4 is shown as an example of the needed changes. We specify a null value for the DATA argument of the FORMAT_CELL method, and then specify the formula using a style override:

```
formula = '=COUNTIF(''Placebo''!D:D, "dead - prostatic ca")';

formula_attr = 'tagattr=' || quote('formula:' || formula);


ODSobj.format_cell(data: '',
                   style_attr: catx(' ', background_color, border_color,
                                    formula_attr) );
```

Similar changes are needed for the cells that provide a count of subjects who died of prostate cancer or heart disease. Table 6 shows the cells that need to be changed, the search range, and the criteria.

Cell Reference	Search Range	Criteria
B4	'Placebo'!D:D	"dead - prostatic ca"
C4	'0.2 mg Estrogen'!D:D	
D4	'1.0 mg Estrogen'!D:D	
E4	'5.0 mg Estrogen'!D:D	
B5	'Placebo'!D:D	"dead - heart or vascular"
C5	'0.2 mg Estrogen'!D:D	
D5	'1.0 mg Estrogen'!D:D	
E5	'5.0 mg Estrogen'!D:D	

Table 6. COUNTIF Search Range and Criteria for Patients That Died of Prostate Cancer or Heart Disease

We use similar code to determine the number of subjects that survived, except we use different criteria in the COUNTIF function. This code counts the number of subjects in the "Placebo" group (cell B9):

```
formula = '=COUNTIF(''Placebo''!D:D, "alive")';

formula_attr = 'tagattr=' || quote('formula:' || formula);

ODSobj.format_cell(data: '',
                   style_attr: catx(' ', background_color, border_color,
                                    formula_attr) );
```

The code for the remaining cells in this row (C9 - E9) needs to be changed. The cells that need to be changed, the search range, and the criteria for these cells are shown in Table 7.

Cell Reference	Search Range	Criteria
B9	'Placebo'!D:D	"alive"
C9	'0.2 mg Estrogen'!D:D	
D9	'1.0 mg Estrogen'!D:D	
E9	'5.0 mg Estrogen'!D:D	

Table 7. COUNTIF Search Range and Criteria for Surviving Patients

A different type of criteria is needed to compute the number of subjects that died of other causes. Refer to row 6 in Figure 1. We count the number that died of any cause, using a wild card for the search criteria ("dead*") to find all cells with text beginning with "dead". Then we subtract the number that died of prostate cancer or heart disease.

The code for cell B6 is shown as an example of the needed changes. We specify a null value for the DATA argument of the FORMAT_CELL method, and then specify the formula using a style override:

```
formula = '=COUNTIF(''Placebo''!D:D, "dead*")-B4-B5';

formula_attr = 'tagattr=' || quote('formula:' || formula);
```

```
ODSobj.format_cell(data: '',
                   style_attr: catx(' ', solid_line, background_color,
                                    border_color, formula_attr) );
```

Information needed to construct code to apply these changes is shown in Table 8.

Cell Reference	Search Range	Criteria	Calculation
B6	'Placebo'!D:D	"dead*"	-B4-B5
C6	'0.2 mg Estrogen'!D:D		-C4-C5
D6	'1.0 mg Estrogen'!D:D		-D4-D5
E6	'5.0 mg Estrogen'!D:D		-E4-E5

Table 8. COUNTIF Search Range and Criteria for Patients That Died of Other Causes

Our output now matches Figure 1. For the final SAS code used to create the output of Figure 1 and Figure 2, see "Appendix: The Final SAS Code".

SAS SERVER TECHNOLOGY

You can deliver dynamically generated SAS output in Microsoft Excel using the Application Dispatcher, the SAS® Stored Process Server, or the SAS® Viya® Compute Server. The Application Dispatcher is part of SAS/IntrNet® software. The SAS Stored Process Server is part of SAS® Integration Technologies and is included with server offerings that use the SAS® Business Analytics infrastructure (for example, SAS® BI Server and SAS® Enterprise BI Server). The SAS Viya Compute server is included with SAS Viya.

These products enable you to execute SAS programs from a Web browser or any other client that can open an HTTP connection. They can run on any platform where SAS is licensed. SAS software does not need to be installed on the client machine.
The SAS programs that you execute from the browser can contain any combination of DATA step, procedure, or macro code. Thus, all the code that has been shown up to this point can be executed by the Application Dispatcher, the SAS Stored Process Server, and the SAS Viya Compute server.

Program execution is typically initiated by accessing a URL that points to the SAS server program. Parameters are passed to the program as name/value pairs in the URL. The SAS server takes these name/value pairs and constructs SAS macro variables that are available to the SAS program.

Figure 10 shows a Web page that can deliver SAS output directly to Microsoft Excel, using a Web browser as the client.

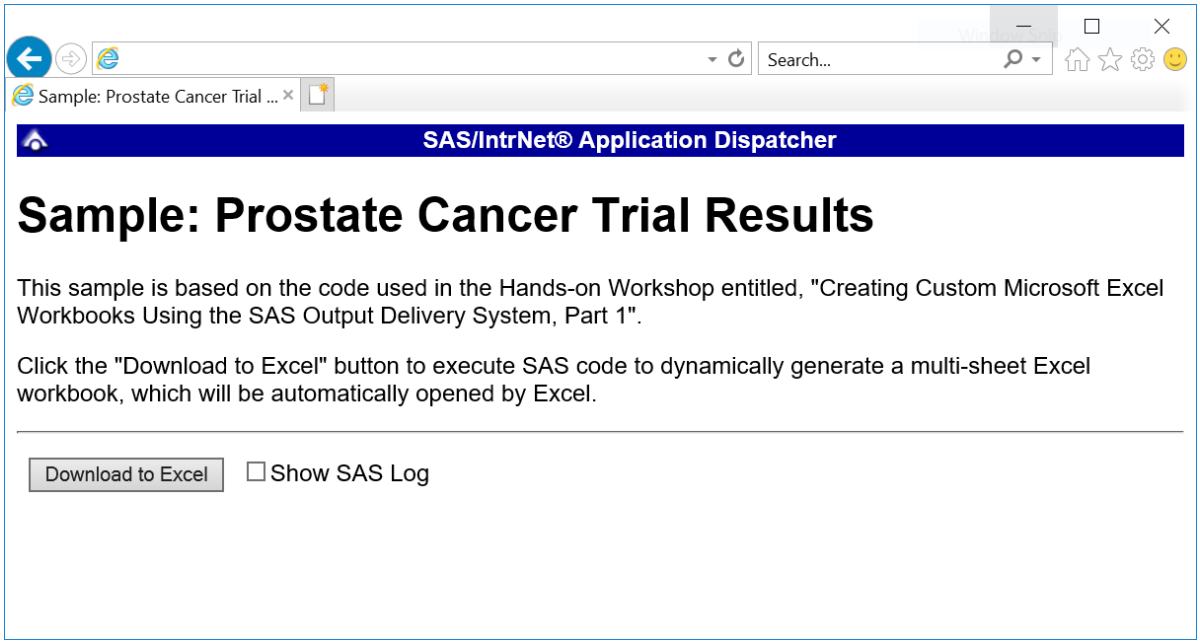


Figure 10. Web Page to Drive a SAS/IntrNet Application

Clicking Download to Excel executes a slightly modified version of the SAS code that we have been working on.

The modifications for SAS/IntrNet and Stored Process applications are as follows:

❶ `%let RV=%sysfunc(appsrv_header(Content-type, application/vnd.ms-excel));`

❷ `%let RV=%sysfunc(appsrv_header(Content-disposition, %str(attachment; filename="ProstateCancer.xlsx")));  * Ignore line wrapping;`

```
ods _all_ close;

ods excel file=_webout style=HTMLBlue ... ;

*  Remainder of the "final" SAS code;

ods excel close;
```

The first MIME header causes the SAS output to be opened by Microsoft Excel, instead of being rendered by the Web browser (❶). This statement is required.

The second MIME header causes the filename to be displayed in the File Download dialog box (❷). As you can see in Figure 11, the filename appears as ProstateCancer.xlsx. This header might cause problems with some versions of Microsoft Excel, so be sure to test your applications before deploying them in a production environment. This statement is optional.

To run your code using the SAS Viya Compute Server, use the SAS Job Execution Web Application to create a job with these input parameters:

_OUTPUT_TYPE	xlsx
_CONTDISP_FILENAME	ProstateCancer

Then specify this code:

```
ods _all_ close;

ods excel file=_webout style=HTMLBlue ... ;

*  Remainder of the "final" SAS code;

ods excel close;
```

In all cases, the reserved _WEBOUT FILEREF is defined by the SAS server and directs output from the SAS server to the client. Modify your existing ODS statement to direct the output to this FILEREF instead of to an external disk file.

When you click the Download to Excel button on the Web page, you are presented with the File Download dialog box (Figure 11). You can then click Open to immediately open your SAS output using Microsoft Excel, or you can click Save.

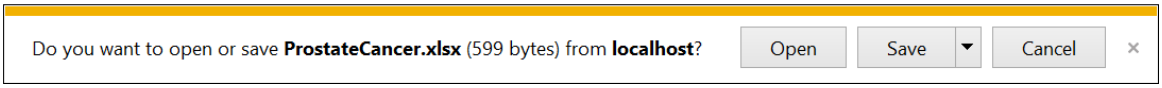


Figure 11. File Download Dialog Box

For more detailed information and other examples, see the SAS/IntrNet Application Dispatcher, SAS Stored Process, and SAS Job Execution Web Application documentation (SAS Institute Inc. 2019c, 2019f, 2019d). This author's earlier papers provide examples of dynamically delivering output to Excel (DelGobbo 2020).

CONCLUSION

The ODS destination for Excel, available in SAS 9.4M3 and later, provides an easy way to export your SAS data to Excel workbooks in the XLSX format. By using ODS Report Writing Interface and style overrides, you can customize the output to achieve your design goals.

APPENDIX: THE FINAL SAS CODE

The final SAS code used to create the output of Figure 1 and Figure 2 follows:

```
ods excel file='directory-location\ProstateCancer.xlsx' style=HTMLBlue
  options(embedded_titles='yes'
          suppress_bylines='yes');

title 'Summary of Outcomes for Placebo and Active Drug';
footnote;

ods excel options(sheet_name='Outcome Summary');

data _null_;

length formula $200 formula_attr $225;

*  Define variables for ODS style attributes;

bold_text        = 'fontweight=bold';
solid_line       = 'bordertopcolor=black';
border_color     = 'bordercolor=#fafbfe';
background_color = 'backgroundcolor=#fafbfe';
```

```
*  Declare the ODS output object;

declare odsout ODSobj();

*  Start the table;

ODSobj.table_start();

*  Column heading row;

ODSobj.row_start();  *  Cells A3 - F3;
  ODSobj.format_cell(data: 'Outcome',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );

  ODSobj.format_cell(data: 'Placebo',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );

  ODSobj.format_cell(data: '0.2 mg',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );

  ODSobj.format_cell(data: '1.0 mg',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );

  ODSobj.format_cell(data: '5.0 mg',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );

  ODSobj.format_cell(data: 'Row Totals',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );
ODSobj.row_end();

*  Data row 1;

ODSobj.row_start();  *  Cells A4 - F4;
  ODSobj.format_cell(data: 'Deceased - PCA',
                     just: 'L',
                     style_attr: catx(' ', background_color,
                                      border_color) );

  formula = '=COUNTIF(''Placebo''!D:D, "dead - prostatic ca")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);

  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''0.2 mg Estrogen''!D:D, "dead - prostatic ca")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''1.0 mg Estrogen''!D:D, "dead - prostatic ca")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
```

```
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''5.0 mg Estrogen''!D:D, "dead - prostatic ca")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  ODSobj.format_cell(data: '=SUM(B4:E4)',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );
  ODSobj.row_end();

*  Data row 2;

ODSobj.row_start();  *  Cells A5 - F5;
  ODSobj.format_cell(data: 'Deceased - Heart or Vascular',
                     just: 'L',
                     style_attr: catx(' ', background_color,
                                      border_color) );

  formula = '=COUNTIF(''Placebo''!D:D, "dead - heart or vascular")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''0.2 mg Estrogen''!D:D, "dead - heart or vascular")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''1.0 mg Estrogen''!D:D, "dead - heart or vascular")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''5.0 mg Estrogen''!D:D, "dead - heart or vascular")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );
  ODSobj.format_cell(data: '=SUM(B5:E5)',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );
ODSobj.row_end();

*  Data row 3;

ODSobj.row_start();  *  Cells A6 - F6;
  ODSobj.format_cell(data: 'Deceased - Other',
                     just: 'L',
                     style_attr: catx(' ', background_color,
                                      border_color) );
```

```
  formula = '=COUNTIF(''Placebo''!D:D, "dead*")-B4-B5';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''0.2 mg Estrogen''!D:D, "dead*")-C4-C5';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''1.0 mg Estrogen''!D:D, "dead*")-D4-D5';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''5.0 mg Estrogen''!D:D, "dead*")-E4-E5';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  ODSobj.format_cell(data: '=SUM(B6:E6)',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );
ODSobj.row_end();

*  Total Deceased row;

ODSobj.row_start();  *  Cells A7 - F7;
  ODSobj.format_cell(data: 'Total Deceased',
                     just: 'L',
                     style_attr: catx(' ', solid_line, background_color,
                                      border_color) );

  ODSobj.format_cell(data: '=SUM(B4:B6)',
                     style_attr: catx(' ', bold_text, solid_line,
                                      background_color, border_color) );

  ODSobj.format_cell(data: '=SUM(C4:C6)',
                     style_attr: catx(' ', bold_text, solid_line,
                                      background_color, border_color) );

  ODSobj.format_cell(data: '=SUM(D4:D6)',
                     style_attr: catx(' ', bold_text, solid_line,
                                      background_color, border_color) );

  ODSobj.format_cell(data: '=SUM(E4:E6)',
                     style_attr: catx(' ', bold_text, solid_line,
                                      background_color, border_color) );

  ODSobj.format_cell(data: '=SUM(B7:E7)',
                     style_attr: catx(' ', bold_text, solid_line,
                                      background_color, border_color) );
ODSobj.row_end();

*  Blank row;
```

```
ODSobj.row_start();  *  Merged Cells A8 - F8;
  ODSobj.format_cell(data: '',
                     column_span: 6,
                     style_attr: catx(' ', background_color,
                                      border_color) );
ODSobj.row_end();

*  Data row 4;

ODSobj.row_start();  *  Cells A9 - F9;
  ODSobj.format_cell(data: 'Survived',
                     just: 'L',
                     style_attr: catx(' ', background_color,
                                      border_color) );

  formula = '=COUNTIF(''Placebo''!D:D, "alive")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''0.2 mg Estrogen''!D:D, "alive")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''1.0 mg Estrogen''!D:D, "alive")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  formula = '=COUNTIF(''5.0 mg Estrogen''!D:D, "alive")';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );

  ODSobj.format_cell(data: '=SUM(B9:E9)',
                     style_attr: catx(' ', bold_text, background_color,
                                      border_color) );
ODSobj.row_end();
*  Blank row;

ODSobj.row_start();  *  Merged Cells A10 - F10;
  ODSobj.format_cell(data: '',
                     column_span: 6,
                     style_attr: catx(' ', background_color,
                                      border_color) );
ODSobj.row_end();

*  Footer row;

ODSobj.row_start();  *  Cell A11, and Merged Cells B11 - F11;
  ODSobj.format_cell(data: '',
                     style_attr: catx(' ', background_color,
                                      border_color) );
```

```
  formula = '=CONCATENATE("Total number of subjects enrolled in the trial
is ", SUM(F7,F9) )';
  formula_attr = 'tagattr=' || quote('formula:' || formula);
  ODSobj.format_cell(data: '',
                     column_span: 5,
                     style_attr: catx(' ', background_color, border_color,
                                      formula_attr) );
ODSobj.row_end();

*  End the table;

ODSobj.table_end();

run;

title 'Detail Data for #byval(rx)';
footnote;

ods excel options(sheet_interval='bygroup'
                  sheet_name='#byval(rx)'
                  autofilter='4-5');

* One worksheet created for each distinct BY value;

proc report data=sample.ProstateCancer nowindows split='*'
    style(column)=[just=center];
  by rx;
  column PatNo Age SZ Status HX EKG;

  define PatNo  / display 'Subject*ID';
  define Age    / display 'Age';
  define SZ     / display 'Size of*Primary Tumor*(cm2)';
  define Status / display style(column)=[just=left];
  define HX     / display 'History of*Cardiovascular*Disease'
                    format=boolean.;
  define EKG    / display 'EKG*Outcome' style(column)=[just=left];

  compute PatNo;
    *  Change background color for alternating rows;
    RowNum+1;
    if (mod(RowNum, 2) eq 0)
    then call define(_row_, 'style', 'style=[background=#acb9ca]');
  endcomp;

  compute Status;
    *  Change background color for certain cells;
    if (Status eq 'dead - heart or vascular')
    then call define('Status', 'style', 'style=[background=#f7a085]');
  endcomp;

  format rx rx.;
run; quit;

ods excel close;
```

REFERENCES

Andrews, D.F. and A.M. Herzberg. 1985. *Data: A Collection of Problems from Many Fields for the Student and Research Worker*. 261-274. New York: Springer-Verlag.

Bailar III, John C. and David P. Byar. 1970: "Estrogen Treatment for Cancer of the Prostate: Early Results with 3 Doses of Diethylstilbestrol and Placebo". *Cancer*. 26(2):257-261.

Byar, D.P. and S.B. Green. 1980: "The Choice of Treatment for Cancer Patients Based on Covariate Information: Application to Prostate Cancer". *Bull Cancer*. 67(4):477-490.

DelGobbo, Vincent. 2011. "Creating Stylish Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS®". *Proceedings of the SAS Global Forum 2011 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/170-2011.pdf

DelGobbo, Vincent. 2020. "Related Papers". Cary, NC: SAS Institute Inc. Available at http://support.sas.com/rnd/papers/intro-multisheet-excel-with-sas/index.html

SAS Institute Inc. 2009. "Sample 36900: Instructions for viewing all of the style templates that are shipped with SAS®". Available at http://support.sas.com/techsup/notes/v8/36/900.html

SAS Institute Inc. 2019a. "#BYVALn | #BYVAL(variable-name)". *SAS® 9.4 Global Statements: Reference*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=lestmtsglobal&docsetTarget=p10gcmrmf83iaxn1ilrx4pra969n.htm#n17bs500gdppsxn1rpsu0nlxr18e

SAS Institute Inc. 2019b. "HEXw. Format". *SAS® 9.4 Formats and Informats: Reference*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=leforinforref&docsetTarget=n0ueabv26pr2fwn19uxk2f4bf10y.htm

SAS Institute Inc. 2019c. "Introduction to the Application Dispatcher". *SAS/IntrNet® 9.4: Application Dispatcher*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?docsetId=dispatch&docsetTarget=p06h82ux8glu1pn16k9dxw8tjpyz.htm&docsetVersion=9.4

SAS Institute Inc. 2019d. "Overview". *SAS® Job Execution Web Application 2.2: User's Guide*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=jobexecug&docsetTarget=n055josnxfatfwn1pyr7p1ah7225.htm

SAS Institute Inc. 2019e. "SAS® 9 Reporting Procedure Styles Tip Sheet". Available at http://support.sas.com/rnd/base/ods/scratch/reporting-styles-tips.pdf

SAS Institute Inc. 2019f. "What Are SAS Stored Processes?" *SAS® 9.4 Stored Processes: Developer's Guide, Third Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=itechcdc&cdcVersion=9.4&docsetId=stpug&docsetTarget=n180km1hadyuton13gx4bzlqhpwr.htm

SAS Institute Inc. 2020a. "CAT Function". *SAS® 9.4 Functions and CALL Routines: Reference, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=lefunctionsref&docsetTarget=p0wism8jqta3qmn1swtpxrs71e9a.htm

SAS Institute Inc. 2020b. "CATX Function". *SAS® 9.4 Functions and CALL Routines: Reference, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=lefunctionsref&docsetTarget=n0p7wxtk0hvn83n1pveisbcp2ae9.htm

SAS Institute Inc. 2020c. "FORMAT_CELL Method". *SAS® 9.4 Output Delivery System: Advanced Topics*, Third Edition. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?docsetId=odsadvug&docsetTarget=p12q0pykppxbz8n1200luexgbz4t.htm&docsetVersion=9.4

SAS Institute Inc. 2020d. "MOD Function". *SAS® 9.4 Functions and CALL Routines: Reference, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=lefunctionsref&docsetTarget=p0ebxby8k7049en12wrfx28vyx4t.htm

SAS Institute Inc. 2020e. "ODS EXCEL Statement". *SAS® 9.4 Output Delivery System: User's Guide, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?docsetId=odsug&docsetTarget=p09n5pw9ol0897n1qe04zeur27rv.htm&docsetVersion=9.4

SAS Institute Inc. 2020f. "QUOTE Function". *SAS® 9.4 Functions and CALL Routines: Reference, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=lefunctionsref&docsetTarget=p059exu866hqw2n0zw9aoxf3p6oj.htm

SAS Institute Inc. 2020g. "Reference and Examples". *SAS® 9.4 Output Delivery System: Advanced Topics*, Third Edition. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?docsetId=odsadvug&docsetTarget=p1sow2hkpzavmun195l2ig8qufz7.htm&docsetVersion=9.4

SAS Institute Inc. 2020h. "Style Attributes Detailed Information". *SAS® 9.4 Output Delivery System: User's Guide, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?docsetId=odsug&docsetTarget=p1pt77toue3iyun0z4l9gth5as9f.htm&docsetVersion=9.4

SAS Institute Inc. 2020i. "Terminology". *SAS® 9.4 Output Delivery System: Advanced Topics*, Third Edition. Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?docsetId=odsadvug&docsetTarget=p0i2uaqbqdpnwqn1ru9iocvdfseg.htm&docsetVersion=9.4

RECOMMENDED READING

Microsoft Corporation. 2020. "Excel functions (alphabetical)". Available at https://support.office.com/en-us/article/excel-functions-alphabetical-b3944572-255d-4efb-bb96-c6d90033e188

SAS Institute Inc. 2019. "Report Procedure". *Base SAS® 9.4 Procedures Guide, Seventh Edition.* Cary, NC: SAS Institute Inc. Available at https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.5&docsetId=proc&docsetTarget=p0bqogcics9o4xn17yvt2qjbgdpi.htm

CONTACT INFORMATION

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

Vincent DelGobbo
SAS Institute Inc.
sasvcd@sas.com

LinkedIn: https://www.linkedin.com/in/vince-delgobbo-a717b88/

Papers and Resources: http://www.sas.com/reg/gen/corp/867226?page=Resources

If your registered in-house or local SAS users group would like to request this presentation as your annual SAS presentation (as a seminar, talk, or workshop) at an upcoming meeting, please submit an online User Group Request Form (bit.ly/2W2KGUB) at least eight weeks in advance.