

## New for SAS<sup>®</sup> 9.4: A Technique for Including Text and Graphics in Your Microsoft Excel Workbooks, Part 1

Vincent DelGobbo, SAS Institute Inc.

### ABSTRACT

A new ODS destination for creating Microsoft Excel workbooks is available starting in the third maintenance release for SAS<sup>®</sup> 9.4. This destination creates native Microsoft Excel XLSX files, supports graphic images, and offers other advantages over the older ExcelXP tagset. In this presentation you learn step-by-step techniques for quickly and easily creating attractive multi-sheet Excel workbooks that contain your SAS<sup>®</sup> output. 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. Using the ExcelXP tagset with earlier versions of SAS to create multi-sheet workbooks is also discussed.

### INTRODUCTION

This paper explains how to use the Excel ODS destination, available in the third maintenance release for Base SAS<sup>®</sup> 9.4 and later, to create the Excel workbook shown in [Figure 1](#). The workbook includes four worksheets that display fictional cholesterol lab results by patient gender for the baseline test and the test results for a given day (visit 5) during a clinical trial.

Notable features of this workbook include the following:

1. Graphic images (scatter plots) are included in the **Female - Plot** and **Male - Plot** worksheets.
2. Worksheet names are customized.
3. Title text is displayed in the document body.
4. Arrow characters are used to indicate the direction of change from the baseline value.
5. Values in the numeric columns are displayed using Excel formats, not SAS formats.
6. The workbook is created in the native Excel XLSX file format without hand-editing.

The REPORT procedure creates the tabular output. The SGPLOT procedure, also part of Base SAS, creates the scatter plots. The shaded rectangle in the scatter plots shows the lower and upper reference range limits for cholesterol lab results, and helps to identify values that are outside of this range.

The code in this paper was tested using the third maintenance release for SAS 9.4 and Microsoft Excel 2010 software.

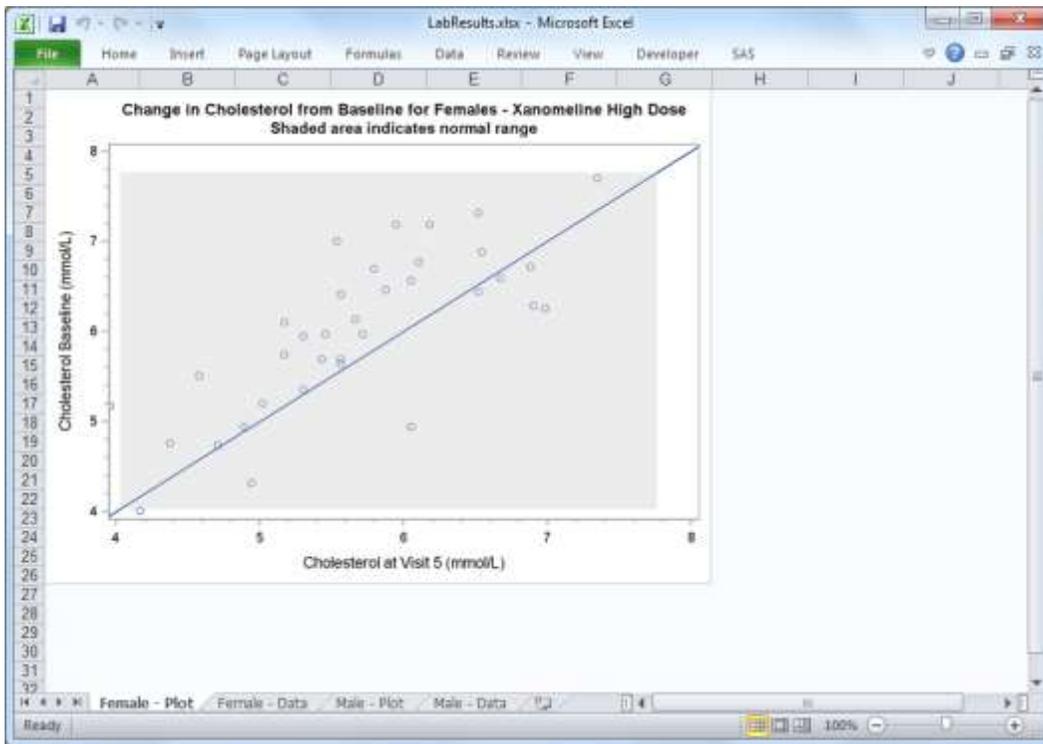
### REQUIREMENTS

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

- The third maintenance release for Base SAS 9.4 and later on any supported operating system (z/OS, UNIX, and so on) and hardware.
- Microsoft Excel 2007 and later.

The Excel destination is available only in the third maintenance release for SAS 9.4 and later. If you have an earlier version of SAS you must use the ExcelXP tagset instead of the Excel ODS destination. The ExcelXP tagset creates a Microsoft Excel XML Spreadsheet file (XML) for use with Excel 2002 and later, and not an XLSX file like the new Excel destination. Instructions for using the old tagset are discussed in each example in this paper. Other papers on the ExcelXP tagset are available from this author (DelGobbo [2015](#)).

You can find information about requesting a maintenance release for SAS at <http://support.sas.com/software/maintenance>.



	Subject Identifier	Age	Baseline Result	Unit	Visit 5 Result	Unit	Change from Baseline	Baseline Date/Time of Specimen Collection	Visit 5 Date/Time of Specimen Collection
4	01-701-1034	77	7.71	mmol/L	7.34	mmol/L	↓	2014-06-24T15:30	2014-07-29T08:40
5	01-701-1133	81	5.35	mmol/L	5.30	mmol/L	↓	2012-10-23T10:57	2012-11-26T11:52
6	01-701-1146	75	6.10	mmol/L	5.17	mmol/L	↓	2013-05-07T13:52	2013-06-16T12:45
7	01-701-1287	56	4.73	mmol/L	4.71	mmol/L	↓	2014-01-17T12:16	2014-02-20T11:09
8	01-701-1383	72	5.51	mmol/L	4.58	mmol/L	↓	2013-01-22T11:43	2013-03-12T11:50
9	01-703-1258	78	5.74	mmol/L	5.17	mmol/L	↓	2012-07-10T16:30	2012-08-17T11:15
10	01-703-1295	88	6.59	mmol/L	6.67	mmol/L	↑	2013-10-24T12:10	2013-12-18T12:45
11	01-703-1335	67	6.13	mmol/L	5.66	mmol/L	↓	2013-12-28T16:10	2014-04-13T10:00
12	01-703-1439	76	7.01	mmol/L	5.53	mmol/L	↓	2014-02-27T10:45	2014-04-08T11:00
13	01-704-1008	76	6.41	mmol/L	5.56	mmol/L	↓	2012-11-25T13:05	2013-02-09T09:10
14	01-704-1074	80	4.32	mmol/L	4.94	mmol/L	↑	2014-01-15T09:00	2014-02-22T13:59
15	01-705-1280	56	6.70	mmol/L	5.79	mmol/L	↓	2014-01-11T14:45	2014-02-13T12:00
16	01-705-1281	73	6.28	mmol/L	6.90	mmol/L	↑	2013-11-26T11:45	2013-12-28T10:25
17	01-705-1310	74	5.17	mmol/L	3.96	mmol/L	↓	2013-10-26T14:15	2013-11-30T09:10
18	01-705-1377	63	4.76	mmol/L	4.37	mmol/L	↓	2013-12-29T16:15	2014-02-07T09:25
19	01-706-1049	60	5.20	mmol/L	5.02	mmol/L	↓	2013-04-30T12:00	2013-06-11T11:00
20	01-708-1178	77	6.44	mmol/L	6.52	mmol/L	↑	2013-12-13T13:45	2014-02-02T10:50
21	01-708-1347	61	7.32	mmol/L	6.52	mmol/L	↓	2013-04-11T11:45	2013-05-16T11:45
22	01-708-1406	71	4.94	mmol/L	6.05	mmol/L	↑	2013-12-06T11:20	2014-01-22T10:55
23	01-709-1099	79	5.69	mmol/L	5.56	mmol/L	↓	2013-10-14T09:50	2013-11-27T11:15
24	01-709-1168	72	5.95	mmol/L	5.30	mmol/L	↓	2013-07-25T09:50	2013-08-30T10:35
25	01-710-1070	85	5.69	mmol/L	5.43	mmol/L	↓	2012-08-28T12:40	2012-10-06T13:50
26	01-710-1137	79	4.01	mmol/L	4.16	mmol/L	↑	2013-10-01T10:45	2013-11-08T12:20

Figure 1. Multi-Sheet Excel Workbook Generated by the Excel ODS Destination

## LIMITATIONS

You can use the Excel destination options with all procedure output, but ODS style overrides apply only to the PRINT, REPORT, and TABULATE procedures. Excel destination options and ODS style overrides are discussed in the "[Understanding and Using the Excel Destination Options](#)" and "[Understanding and Using ODS Style Overrides](#)" sections, respectively.

You cannot use the techniques described in this paper to update existing workbooks. ODS creates the entire document on each execution, and cannot alter existing workbooks.

You must specify the FILESYSTEM=HFS SAS system option on z/OS because the Excel ODS destination works only with the HFS file system (SAS Institute Inc. [2015b](#)). This option is not required if you are using the older ExcelXP tagset.

## IMPORTANT NOTE

By default, the Excel ODS destination in the third maintenance release for SAS 9.4 creates a workbook with all worksheets grouped (Figure 2), and any changes made to one worksheet are made to **every worksheet**. To ungroup the worksheets, right-click on any worksheet, click **Ungroup Sheets** in the pop-up menu, and then save the file. The **[Group]** text is not displayed in the Windows title bar after the sheets are ungrouped. Worksheets will not be grouped by default in a future release of SAS.



**Figure 2. Excel Title Bar Indicating Grouped Worksheets**

## SAMPLE DATA

Table 1 presents the column properties for the BASELINE\_VISIT5 SAS table that is used to create the Excel workbook shown in [Figure 1](#). Cholesterol test results for the baseline visit and the fifth visit are included in this table.

Column Name	Description	Typical Values
usubjid	Character subject identifier	01-701-1034, 01-717-1109
age	Patient age	56, 74, 88
sex	Patient gender	F, M
lbtst	Laboratory test name	PH, CHOLESTEROL, GLUCOSE
baseline_lbstresn	Baseline laboratory result in standard units	3.52, 4.01, 6.00
baseline_lbstresu	Baseline laboratory result units	mmol/L
visit5_lbstresn	Visit 5 laboratory result in standard units	3.39, 5.30, 7.01
visit5_lbstresu	Visit 5 laboratory result units	mmol/L
baseline_lbdtc	Character date/time of baseline specimen collection	2014-01-15T09:00, 2013-01-09T14:57
visit5_lbdtc	Character date/time of visit 5 specimen collection	2014-04-13T10:00, 2012-10-06T13:50

**Table 1. Column Properties and Representative Data Values for the BASELINE\_VISIT5 SAS Table**

Below, you can see the annotation data sets that are used with the SGPLOT procedure to control the size and appearance of the shaded rectangle in the scatter plots. The properties of these data sets are presented in [Table 2](#) for female patients and [Table 3](#) for male patients. The values of X1, Y1, WIDTH, and HEIGHT control the size of the shaded rectangle representing the lower and upper reference range limits for cholesterol test results.

Column Name	Description	Value
function	Drawing action to perform	rectangle
x1	X coordinate of the bottom, left corner	4.03
y1	Y coordinate of the bottom, left corner	4.03
width	Width of the annotation	3.73
height	Height of the annotation	3.73
anchor	Anchor point of the annotation	bottomleft
display	Specifies the fill and outline properties	fill
fillcolor	Fill color for the annotation	grayc0
filltransparency	Transparency for the fill portion of an annotation	0.7
widthunit	Dimension unit to use for the WIDTH value	data
heightunit	Dimension unit to use for the HEIGHT value	data
x1space	Drawing space of the annotation's X coordinate	datavalue
y1space	Drawing space of the annotation's Y coordinate	datavalue

**Table 2. Column Properties for the F\_CHOLESTEROL\_ANNO SAS Table**

Column Name	Description	Value
function	Drawing action to perform	rectangle
x1	X coordinate of the bottom, left corner	3.85
y1	Y coordinate of the bottom, left corner	3.85
width	Width of the annotation	3.55
height	Height of the annotation	3.55
anchor	Anchor point of the annotation	bottomleft
display	Specifies the fill and outline properties	fill
fillcolor	Fill color for the annotation	grayc0
filltransparency	Transparency for the fill portion of an annotation	0.7
widthunit	Dimension unit to use for the WIDTH value	data
heightunit	Dimension unit to use for the HEIGHT value	data
x1space	Drawing space of the annotation's X coordinate	datavalue
y1space	Drawing space of the annotation's Y coordinate	datavalue

**Table 3. Column Properties for the M\_CHOLESTEROL\_ANNO SAS Table**

## 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 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. In this paper, we use an ODS destination named "Excel" to create an Excel workbook in XLSX format that has multiple worksheets.

The Excel workbook in [Figure 1](#) was created using the Excel ODS destination and the HTMLBlue ODS style supplied by SAS. 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 ODS.

Here are the general ODS statements to generate XLSX output that is compatible with Excel 2007 and later:

- ❶ `ods _all_ close;`
- ❷ `ods Excel file='file-name.xlsx' style=style-name ... ;`  
`* Your SAS procedure code here;`
- ❸ `ods Excel close;`

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

The second ODS statement (❷) uses the Excel destination to generate the XLSX output and then store the output in a file. 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, click the **Full Code** tab in SAS Sample 36900 (SAS Institute Inc. [2009](#)).

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

**Note:** If you place the files where users can access them over a network, you should set file permissions to prevent accidental alteration.

If you are using SAS 9.1.3 SP4 through SAS 9.4 M2, you must instead use the ExcelXP tagset to create an XML output file that, when opened by Excel, is rendered as a multi-sheet workbook.

Here are the general ODS statements to generate XML output that is compatible with Excel 2002 and later:

```
ods _all_ close;

ods tagsets.ExcelXP file='file-name.xml' style=style-name ... ;
* Your SAS procedure code here;
ods tagsets.ExcelXP close;
```

If you are using the ExcelXP tagset you must also use the XML extension instead of XLS or XLSX because Excel 2007 and later display a warning if the XML extension is not used (Microsoft Corporation [2016a](#)).

Most of the techniques and options discussed in this paper work with both the Excel destination and the tagsets.ExcelXP destination, except that the ExcelXP tagset does not support images. Because the ExcelXP tagset creates files that conform to the Microsoft XML Spreadsheet Specification, and the Microsoft XML Spreadsheet Specification does not support images, the output from graphics procedures cannot be used (Microsoft Corporation [2001](#)).

Other differences between the Excel and tagsets.ExcelXP destinations are discussed in the remaining sections.

## OPENING THE OUTPUT WITH EXCEL

Follow these steps to open the file with Excel:

1. From most versions of Excel, select **File** ➔ **Open**.
2. Navigate to the file or enter the path and filename in the **File name** field.
3. Click **Open** to import the XLSX or XML file.

You can instead navigate to the file using Microsoft Windows Explorer, and then double-click the file to open it with Excel.

**ExcelXP tagset users:** Excel reads and converts the XML file to the Excel format. You can now perform any Excel function on the data. To save a copy of the file in Excel binary (XLS) format, use "Save As" and then select **Microsoft Excel Workbook (\*.xls)** or **Excel 97-2003 Workbook (\*.xls)** from the **Save as type** drop-down list. If you're using Excel 2007 and later and want to save the document in the Microsoft Office Open XML (XLSX) format, then select **Excel Workbook (\*.xlsx)**.

## USING ODS TO CREATE THE MULTI-SHEET EXCEL WORKBOOK

Here is a listing of the **basic** SAS code used to create the Excel workbook:

```
ods _all_ close;

❶ ods Excel file='LabResults.xlsx' style=HTMLBlue;

* First worksheet;

title1 'Change in Cholesterol from Baseline for Females - Xanomeline High
      Dose';
title2 'Shaded area indicates normal range';

❷ proc sgplot data=sample.Baseline_Visit5
      sganno=sample.F_Cholesterol_Anno
      noautolegend;
  where (sex eq 'F' and lbtest eq 'CHOLESTEROL');
  scatter x=visit5_lbstresn y=baseline_lbstresn;
❸ xaxis values=(4 to 8 by 1) minor minorcount=4;
  yaxis values=(4 to 8 by 1) minor minorcount=4;
  lineparm x=4 y=4 slope=1;
  label visit5_lbstresn = 'Cholesterol at Visit 5 (mmol/L)'
      baseline_lbstresn = 'Cholesterol Baseline (mmol/L)';
run; quit;

* Second worksheet;

title2;

❹ proc report data=sample.Baseline_Visit5 nowd split='*';
  where (sex eq 'F' and lbtest eq 'CHOLESTEROL');
  column usubjid age baseline_lbstresn baseline_lbstresu
      visit5_lbstresn visit5_lbstresu change
      baseline_lbdtc visit5_lbdtc;
  define usubjid          / order      'Subject*Identifier';
  define age              / display    '*Age';
  define baseline_lbstresn / display  'Baseline*Result' format=5.2;
  define baseline_lbstresu / display  'Unit';
❺ define visit5_lbstresn  / display  'Visit 5*Result' format=5.2;
  define visit5_lbstresu  / display  'Unit';
  define change           / computed  'Change from*Baseline';
  define baseline_lbdtc   / display   'Baseline Date/Time of*Specimen
      Collection';
  define visit5_lbdtc     / display   'Visit 5 Date/Time of*Specimen
      Collection';

❻ compute change / character length=1;
  * Placeholder for change from baseline indicator;
endcomp;

run; quit;
```

7 \* Third worksheet;

```
title 'Change in Cholesterol from Baseline for Males - Xanomeline High
      Dose';
title2 'Shaded area indicates normal range';

proc sgplot data=sample.Baseline_Visit5
            sganno=sample.M_Cholesterol_Anno
            noautolegend;
  where (sex eq 'M' and lbtest eq 'CHOLESTEROL');
  scatter x=visit5_lbstresn y=baseline_lbstresn;
  xaxis values=(3 to 9 by 1) minorcount=4;
  yaxis values=(3 to 9 by 1) minorcount=4;
  lineparm x=3 y=3 slope=1;
  label visit5_lbstresn = 'Cholesterol at Visit 5 (mmol/L)'
        baseline_lbstresn = 'Cholesterol Baseline (mmol/L)';
run; quit;
```

8 \* Fourth worksheet;

```
title2;

proc report data=sample.Baseline_Visit5 nowd split='*';
  where (sex eq 'M' and lbtest eq 'CHOLESTEROL');
  column usubjid age baseline_lbstresn baseline_lbstresu
         visit5_lbstresn visit5_lbstresu change
         baseline_lbdtc visit5_lbdtc;
  define usubjid          / order      'Subject*Identifier';
  define age              / display    '*Age';
  define baseline_lbstresn / display  'Baseline*Result' format=5.2;
  define baseline_lbstresu / display  'Unit';
  define visit5_lbstresn  / display    'Visit 5*Result' format=5.2;
  define visit5_lbstresu  / display    'Unit';
  define change           / computed   'Change from*Baseline';
  define baseline_lbdtc   / display    'Baseline Date/Time of*Specimen
                                   Collection';
  define visit5_lbdtc     / display    'Visit 5 Date/Time of*Specimen
                                   Collection';

  compute change / character length=1;
  * Placeholder for change from baseline indicator;
endcomp;

run; quit;
```

9 ods Excel close;

As you can see in the ODS statement (9), the Excel destination generates the output and the HTMLBlue style controls the appearance of the output. By default, the Excel destination creates a new worksheet when a SAS procedure creates a new graphic image or new tabular output. Each instance of the SGLOT procedure creates one image and the REPORT procedure creates one table per instance, resulting in an Excel workbook with four worksheets.

The SGLOT procedure (7) uses the F\_CHOLESTEROL\_ANNO table (Table 2) to create the shaded rectangle showing the lower and upper reference range limits for cholesterol test results. A WHERE statement subsets the data based on gender and includes only cholesterol test results.

The remaining SGPLOT procedure statements (❸) control the structure and appearance of the scatter plot. The baseline cholesterol results (BASELINE\_LBSTRESN) are plotted on the y-axis against the results of visit 5 (VISIT5\_LBSTRESN), and a diagonal line of equivalence is drawn using the LINEPARM statement. The line of equivalence helps to quickly identify trends in the data. For example, [Figure 1](#) shows that a high dose of Xanomeline results in a reduction in cholesterol at visit 5 for many female patients.

The PROC REPORT code (❹) creates the tabular results in [Figure 1](#). The COLUMN statement specifies the order to display the columns and the column roles and appearance characteristics are specified in the DEFINE statements (❺). The role of the CHANGE column is specified as **computed**, and the computation is performed in the COMPUTE block (❻).

Similar SGPLOT and REPORT procedure code (❼ and ❽) creates the worksheets for the male patients. The last ODS statement (❾) closes the Excel destination and releases the XLSX file so that it can be opened with Excel.

**ExcelXP tagset users:** To use the ExcelXP tagset, change **Excel** to **tagsets.ExcelXP** in statements ❶ and ❾, and use **.xml** for the file extension in statement ❶. If you're using an earlier version of Base SAS that does not include the HTMLBlue style, use another style in statement ❶, such as Printer. Because the ExcelXP tagset does not support graphic images, you should omit the SGPLOT code here and in all future code to avoid creating blank worksheets.

[Figure 3](#) displays the result of executing the basic SAS code, and then opening the LabResults.xlsx file with Excel. Notice that [Figure 3](#) does not match [Figure 1](#). The following problems are in [Figure 3](#):

1. Unattractive, default worksheet names.
2. Missing title text on the tabular output.
3. Incorrect text justification for the data values.
4. Blank "Change from Baseline" column.

In the remaining sections we change the basic SAS code to correct these problems. The complete SAS code used to create the workbook shown in [Figure 1](#) is listed in the section "[The Final SAS Code](#)".

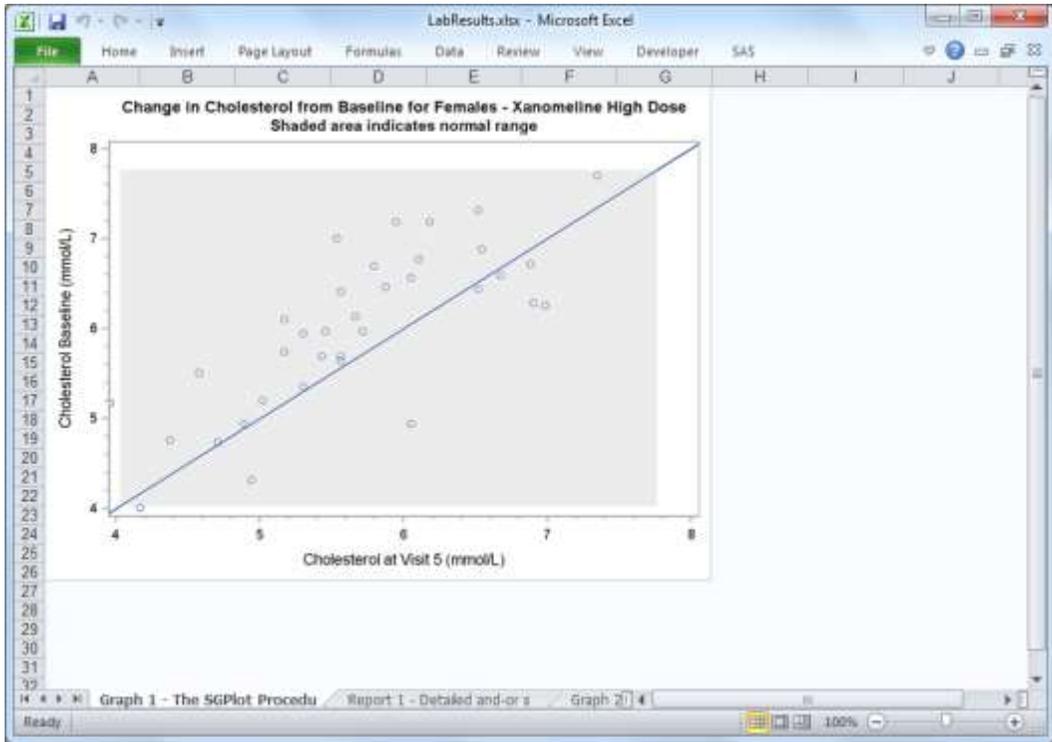
## UNDERSTANDING AND USING THE EXCEL DESTINATION OPTIONS

The Excel destination supports many options that control both the appearance and functionality of the Excel 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.



	A	B	C	D	E	F	G	H	I
1	Subject Identifier	Age	Baseline Result	Unit	Visit 5 Result	Unit	Change from Baseline	Baseline Date/Time of Specimen Collection	Visit 5 Date/Time of Specimen Collection
2	01-701-1034	77	7.71	mmol/L	7.34	mmol/L		2014-06-24T15:30	2014-07-29T08:40
3	01-701-1133	81	5.35	mmol/L	5.30	mmol/L		2012-10-23T10:57	2012-11-26T11:52
4	01-701-1146	75	6.10	mmol/L	5.17	mmol/L		2013-05-07T13:52	2013-06-16T12:45
5	01-701-1287	56	4.73	mmol/L	4.71	mmol/L		2014-01-17T12:16	2014-02-20T11:09
6	01-701-1383	72	5.51	mmol/L	4.58	mmol/L		2013-01-22T11:43	2013-03-12T11:50
7	01-703-1258	78	5.74	mmol/L	5.17	mmol/L		2012-07-10T16:30	2012-08-17T11:15
8	01-703-1295	88	6.59	mmol/L	6.67	mmol/L		2013-10-24T12:10	2013-12-18T12:45
9	01-703-1335	67	6.13	mmol/L	5.66	mmol/L		2013-12-28T16:10	2014-04-13T10:00
10	01-703-1439	76	7.01	mmol/L	5.53	mmol/L		2014-02-27T10:45	2014-04-08T11:00
11	01-704-1008	76	6.41	mmol/L	5.56	mmol/L		2012-11-25T13:05	2013-02-09T09:10
12	01-704-1074	80	4.32	mmol/L	4.94	mmol/L		2014-01-15T09:00	2014-02-22T13:59
13	01-705-1280	56	6.70	mmol/L	5.79	mmol/L		2014-01-11T14:45	2014-02-13T12:00
14	01-705-1281	73	6.28	mmol/L	6.90	mmol/L		2013-11-26T11:45	2013-12-28T10:25
15	01-705-1310	74	5.17	mmol/L	3.96	mmol/L		2013-10-26T14:15	2013-11-30T09:10
16	01-705-1377	63	4.76	mmol/L	4.37	mmol/L		2013-12-29T16:15	2014-02-07T09:25
17	01-706-1049	60	5.20	mmol/L	5.02	mmol/L		2013-04-30T12:00	2013-06-11T11:00
18	01-708-1178	77	6.44	mmol/L	6.52	mmol/L		2013-12-13T13:45	2014-02-02T10:50
19	01-708-1347	61	7.32	mmol/L	6.52	mmol/L		2013-04-11T11:45	2013-05-16T11:45
20	01-708-1406	71	4.94	mmol/L	6.05	mmol/L		2013-12-06T11:20	2014-01-22T10:55
21	01-709-1099	79	5.69	mmol/L	5.56	mmol/L		2013-10-14T09:50	2013-11-27T11:15
22	01-709-1168	72	5.95	mmol/L	5.30	mmol/L		2013-07-25T09:50	2013-08-30T10:35
23	01-710-1070	85	5.69	mmol/L	5.43	mmol/L		2012-08-28T12:40	2012-10-06T13:50
24	01-710-1137	79	4.01	mmol/L	4.16	mmol/L		2013-10-03T10:45	2013-11-08T12:20
25	01-710-1142	76	7.19	mmol/L	6.18	mmol/L		2012-09-12T16:50	2012-10-23T15:20
26	01-710-1187	78	6.57	mmol/L	6.05	mmol/L		2012-10-26T11:30	2012-12-09T09:50
27	01-711-1012	67	5.64	mmol/L	5.56	mmol/L		2013-03-20T14:12	2013-05-02T09:40

Figure 3. Initial Multi-Sheet Excel Workbook Generated by the Excel ODS Destination

For example:

```
ods Excel file='file-name.xlsx' style=style-name ... ;
ods Excel options(option-name='some-value');
* Some SAS procedure code here;
ods Excel options(option-name='default-value');
* Other SAS procedure code here;
ods Excel close;
```

When specifying multiple ODS statements as shown above, specify the FILE, STYLE, or any other keyword or option that is supported by ODS only in the initial ODS statement. Refer to the ODS documentation for a list of available options (SAS Institute Inc. [2015c](#)).

All of the options and code work with any SAS procedure. However, ODS style overrides, discussed in the "[Understanding and Using ODS Style Overrides](#)" section, work only with the PRINT, REPORT, and TABULATE procedures.

**ExcelXP tagset users:** In the preceding ODS statements, specify **tagsets.ExcelXP** instead of **Excel**, and use **.xml** for the file extension instead of **.xlsx**. To see a listing of the supported options printed to the SAS log, submit the following SAS code:

```
filename temp temp;

ods Excel file=temp options(doc='help');
ods Excel close;

filename temp clear;
```

## SPECIFYING WORKSHEET NAMES

ODS generates a unique name for each worksheet, as required by Excel. [Figure 3](#) shows the worksheet names that result from running the initial SAS code.

Use the SHEET\_NAME option to specify a worksheet name. Recall that ODS options remain in effect until the Excel destination is closed. We specify the option four times because we want a different name for each worksheet. The results are shown in [Figure 4](#).

```
ods Excel file='LabResults.xlsx' style=HTMLBlue;

* First worksheet;
ods Excel options(sheet_name='Female - Plot');

title1 ... ; title2 ... ;

proc sgplot ... ; run; quit;

* Second worksheet;
ods Excel options(sheet_name='Female - Report');

title2;

proc report ... ; run; quit;

* Third worksheet;
ods Excel options(sheet_name='Male - Plot');

title1 ... ; title2 ... ;
```

```

proc sgplot ... ; run; quit;

* Fourth worksheet;
ods Excel options(sheet_name='Male - Report');

title2;

proc report ... ; run; quit;

ods Excel close;

```

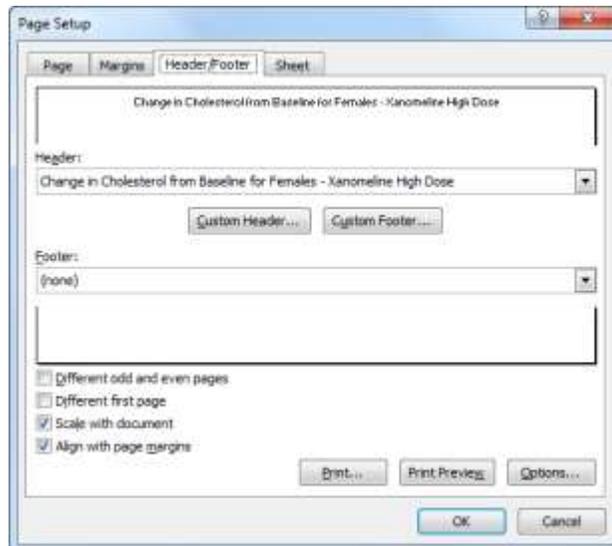


**Figure 4. ODS-Generated Excel Workbook with New Worksheet Names**

**ExcelXP tagset users:** Specify **tagsets.ExcelXP** instead of **Excel**, and use **.xml** for the file extension instead of **.xlsx** in the preceding ODS statements.

### INCLUDING TITLE TEXT IN THE WORKSHEET BODY

By default, SAS titles appear as Excel print headers, which are displayed when the Excel document is printed. You can confirm this by viewing the Excel **Header/Footer** tab in the Page Setup dialog box, shown in Figure 5.



**Figure 5. Excel Page Setup Dialog Box Showing Title Text in the Print Header**

To include title text on-screen, in the worksheet body, use the **EMBEDDED\_TITLES** option:

```

ods Excel file='LabResults.xlsx' style=HTMLBlue;

* Option that affects all worksheets;
ods Excel options(embedded_titles='yes');

* First worksheet;
ods Excel options(sheet_name='Female - Plot');
... ;

```

```

* Second worksheet;
ods Excel options(sheet_name='Female - Report');
... ;

* Third worksheet;
ods Excel options(sheet_name='Male - Plot');
... ;

* Fourth worksheet;
ods Excel options(sheet_name='Male - Report');
... ;

ods Excel close;

```

Because the EMBEDDED\_TITLES option is placed at the beginning of the code and is not changed later, it affects all worksheets. The title text is now included in the worksheet body (Figure 6).

Subject Identifier	Age	Baseline Result	Unit	Visit 5 Result	Unit	Change from Baseline	Baseline Date/Time of Specimen Collection	Visit 5 Date/Time of Specimen Collection
01-701-1034	77	7.71	mmol/L	7.34	mmol/L		2014-06-24T15:30	2014-07-29T08:40
01-701-1133	81	5.35	mmol/L	5.30	mmol/L		2012-10-23T10:57	2012-11-26T11:52
01-701-1146	75	6.10	mmol/L	5.17	mmol/L		2013-05-07T13:52	2013-06-16T12:45
01-701-1287	56	4.73	mmol/L	4.71	mmol/L		2014-01-17T12:16	2014-02-20T11:09
01-701-1383	72	5.51	mmol/L	4.58	mmol/L		2013-01-22T11:43	2013-03-12T11:50
01-703-1258	78	5.74	mmol/L	5.17	mmol/L		2012-07-10T16:30	2012-08-17T11:15
01-703-1295	88	6.59	mmol/L	6.67	mmol/L		2013-10-24T12:10	2013-12-18T12:45
01-703-1335	67	6.13	mmol/L	5.66	mmol/L		2013-12-28T16:10	2014-04-13T10:00
01-703-1439	76	7.01	mmol/L	5.53	mmol/L		2014-02-27T10:45	2014-04-08T11:00
01-704-1008	76	6.41	mmol/L	5.56	mmol/L		2012-11-25T13:05	2013-02-09T09:10
01-704-1074	80	4.32	mmol/L	4.94	mmol/L		2014-01-15T09:00	2014-02-22T13:59
01-705-1280	56	6.70	mmol/L	5.79	mmol/L		2014-01-11T14:45	2014-02-13T12:00
01-705-1281	73	5.28	mmol/L	6.90	mmol/L		2013-11-26T11:45	2013-12-28T10:25
01-705-1310	74	5.17	mmol/L	3.95	mmol/L		2013-10-26T14:15	2013-11-30T09:10
01-705-1377	63	4.76	mmol/L	4.37	mmol/L		2013-12-29T16:15	2014-02-07T09:25
01-706-1049	60	5.20	mmol/L	5.02	mmol/L		2013-04-30T12:00	2013-06-11T11:00
01-708-1178	77	6.44	mmol/L	6.52	mmol/L		2013-12-13T13:45	2014-02-02T10:50
01-708-1347	61	7.32	mmol/L	6.52	mmol/L		2013-04-11T11:45	2013-05-16T11:45
01-708-1406	71	4.94	mmol/L	6.05	mmol/L		2013-12-06T11:20	2014-01-22T10:55
01-709-1099	79	5.69	mmol/L	5.56	mmol/L		2013-10-14T09:50	2013-11-27T11:15
01-709-1168	72	5.95	mmol/L	5.30	mmol/L		2013-07-25T09:50	2013-08-30T10:35
01-710-1070	85	5.69	mmol/L	5.43	mmol/L		2012-08-28T12:40	2012-10-06T13:50
01-710-1117	79	4.01	mmol/L	4.15	mmol/L		2013-10-07T10:45	2013-11-08T12:20

Figure 6. ODS-Generated Excel Workbook with Embedded Title Text

**ExcelXP tagset users:** In the preceding ODS statements, specify **tagsets.ExcelXP** instead of **Excel**, and use **.xml** for the file extension instead of **.xlsx**.

## UNDERSTANDING AND USING ODS STYLE OVERRIDES

You can alter the attributes or style elements used by specific parts of your SAS output by using style overrides. These specific parts of your SAS output are called *locations*. Figure 7 shows the locations that are pertinent to the REPORT procedure output (SAS Institute Inc. 2008). The COLUMN location controls the appearance of data cells.

Style overrides are supported by the PRINT, REPORT, and TABULATE procedures, and can be specified several ways. Here are the two most common formats:

- ❶ `style(location)=[style-attribute-name1=value1  
style-attribute-name2=value2 ...]`
- ❷ `style(location)=style-element-name`

The first format (❶) uses individual style attributes defined inline. For example, this PROC REPORT code alters three attributes of the COLUMN location (data cells) for the MYVAR column:

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

Although this is the most commonly used format, it has some disadvantages. To use the same style override for different variables, you must apply it in multiple places, making your SAS code harder to read and maintain. And, if you want to use the style overrides in other SAS programs, you must copy the list of attribute name/value pairs to the new code. Because of these drawbacks, inline style overrides should be used sparingly.

The second format (❷) overcomes these problems by referencing a style element. Using this format involves creating a new style element, setting the style attributes within the element, and then using the *style element name* in your style override. This results in code that is easier to read, maintain, and reuse. Earlier papers by this author provide a detailed discussion of this topic (DelGobbo 2008, 2009, 2010, 2011).

Refer to the ODS documentation for a full listing of style attributes (SAS Institute Inc. 2015f).

report		
header	header	header
column	column	column
column	column	column
summary	summary	summary
lines		
column	column	column
column	column	column
summary	summary	summary
lines		
lines		

Figure 7. Style Locations for the REPORT Procedure

## CHANGING THE TEXT JUSTIFICATION OF THE DATA CELLS

Some of the data values in the PROC REPORT output are left-justified and others are right-justified. We override the value of the JUST attribute to center the values, as shown in [Figure 1](#).

```
* First worksheet;
... ;

* Second worksheet;
... ;

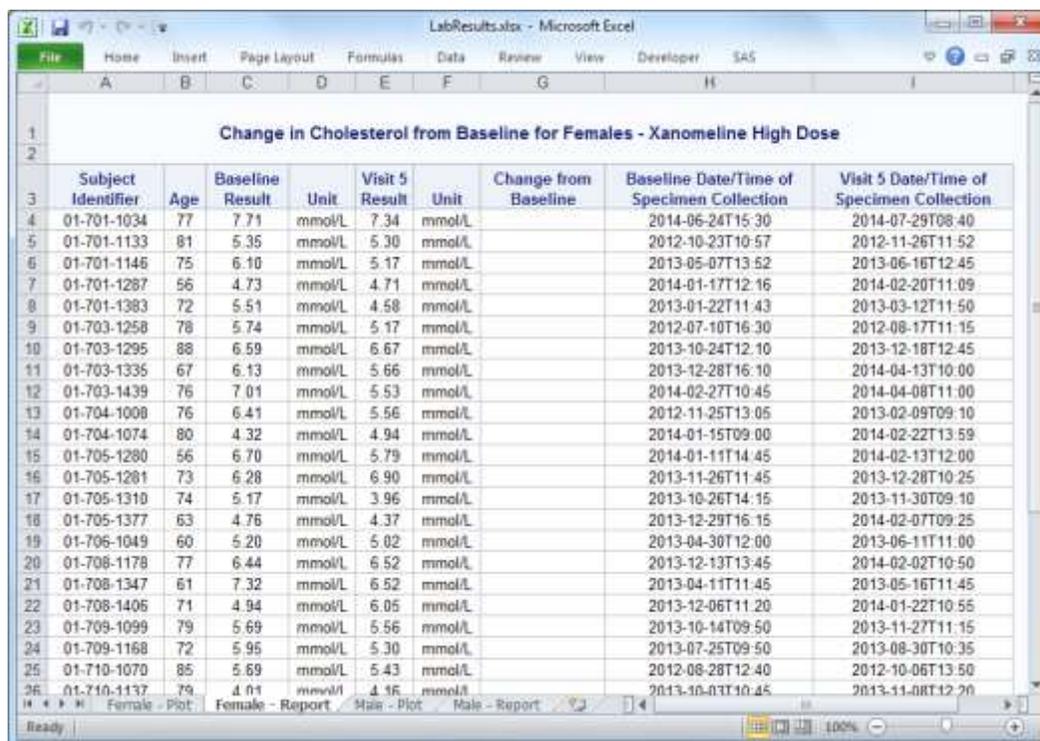
proc report data=sample.Baseline_Visit5 nowd split='*'
             style(column)=[just=center];
  where (sex eq 'F' and lbtest eq 'CHOLESTEROL');
... ;

* Third worksheet;
... ;

* Fourth worksheet;
... ;

proc report data=sample.Baseline_Visit5 nowd split='*'
             style(column)=[just=center];
  where (sex eq 'M' and lbtest eq 'CHOLESTEROL');
... ;
```

Note that no changes were made to the ODS Excel destination options that were previously specified. Specifying the style override in the PROC statement is a convenient way to control the appearance of all columns. The results are shown in [Figure 8](#).



	Subject Identifier	Age	Baseline Result	Unit	Visit 5 Result	Unit	Change from Baseline	Baseline Date/Time of Specimen Collection	Visit 5 Date/Time of Specimen Collection
4	01-701-1034	77	7.71	mmol/L	7.34	mmol/L		2014-06-24T15:30	2014-07-29T08:40
5	01-701-1133	81	5.35	mmol/L	5.30	mmol/L		2012-10-23T10:57	2012-11-26T11:52
6	01-701-1145	75	6.10	mmol/L	5.17	mmol/L		2013-05-07T13:52	2013-06-16T12:45
7	01-701-1287	56	4.73	mmol/L	4.71	mmol/L		2014-01-17T12:16	2014-02-20T11:09
8	01-701-1383	72	5.51	mmol/L	4.58	mmol/L		2013-01-22T11:43	2013-03-12T11:50
9	01-703-1258	78	5.74	mmol/L	5.17	mmol/L		2012-07-10T16:30	2012-08-17T11:15
10	01-703-1295	88	6.59	mmol/L	6.67	mmol/L		2013-10-24T12:10	2013-12-18T12:45
11	01-703-1335	67	6.13	mmol/L	5.66	mmol/L		2013-12-28T16:10	2014-04-13T10:00
12	01-703-1439	76	7.01	mmol/L	5.53	mmol/L		2014-02-27T10:45	2014-04-08T11:00
13	01-704-1008	76	6.41	mmol/L	5.56	mmol/L		2012-11-25T13:05	2013-02-09T09:10
14	01-704-1074	80	4.32	mmol/L	4.94	mmol/L		2014-01-15T09:00	2014-02-22T13:59
15	01-705-1280	56	6.70	mmol/L	5.79	mmol/L		2014-01-11T14:45	2014-02-13T12:00
16	01-705-1281	73	6.28	mmol/L	6.90	mmol/L		2013-11-26T11:45	2013-12-26T10:25
17	01-705-1310	74	5.17	mmol/L	3.96	mmol/L		2013-10-26T14:15	2013-11-30T09:10
18	01-705-1377	63	4.76	mmol/L	4.37	mmol/L		2013-12-29T16:15	2014-02-07T09:25
19	01-706-1049	60	5.20	mmol/L	5.02	mmol/L		2013-04-30T12:00	2013-06-11T11:00
20	01-708-1178	77	6.44	mmol/L	6.52	mmol/L		2013-12-13T13:45	2014-02-02T10:50
21	01-708-1347	61	7.32	mmol/L	6.52	mmol/L		2013-04-11T11:45	2013-05-16T11:45
22	01-708-1406	71	4.94	mmol/L	6.05	mmol/L		2013-12-06T11:20	2014-01-22T10:55
23	01-709-1099	79	5.69	mmol/L	5.56	mmol/L		2013-10-14T09:50	2013-11-27T11:15
24	01-709-1168	72	5.95	mmol/L	5.30	mmol/L		2013-07-25T09:50	2013-08-30T10:35
25	01-710-1070	85	5.69	mmol/L	5.43	mmol/L		2012-08-28T12:40	2012-10-06T13:50
26	01-710-1137	79	4.01	mmol/L	4.95	mmol/L		2013-10-03T10:45	2013-11-08T12:20

Figure 8. ODS-Generated Excel Workbook with Centered Data Values

**ExcelXP tagset users:** Be sure to use the **tagsets.ExcelXP** destination and **.xml** for the file extension. No other code changes are needed.

### COMPUTING THE VALUE FOR THE CHANGE FROM BASELINE COLUMN

We use an up arrow in the CHANGE column to indicate that a value at visit 5 has risen from the baseline value and a down arrow for a value that has declined. The arrow characters are available in the Microsoft Windows Wingdings font.

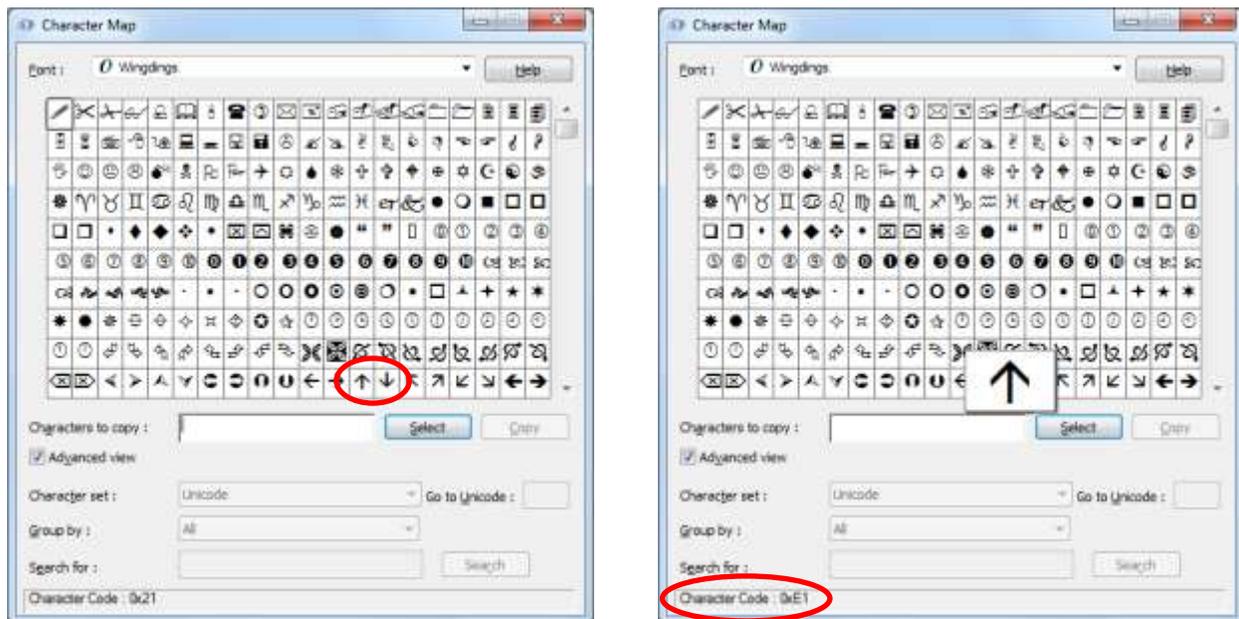
The HTMLBlue ODS style that we're using applies the Arial font to data cells, but we want to use a different font, so we use a style override to change the font to Wingdings. By applying the style override in the DEFINE statement instead of in the PROC statement, we change the font for only the data values in the CHANGE column:

```
define change / computed 'Change from*Baseline'  
                style (column)=[fontfamily=Wingdings];
```

We then use the Windows Character Map to determine the hexadecimal character codes for the up and down arrows:

1. Click the Windows Start button (🌐).
2. In the **Search programs and files** textbox, type **charmap.exe** and then press **Enter**.
3. From the **Font** drop-down list, select **Wingdings**.

The arrow characters are shown at the bottom of the Character Map dialog box (Figure 9, left).



**Figure 9. Windows Character Map Showing Arrow Characters in the Wingdings Font**

Click the up arrow to select it. The hexadecimal character code E1 is displayed at the bottom of the dialog box (Figure 9, right). The code for the down arrow is E2, and we use these two values in both of the COMPUTE blocks.

Here is the code for one COMPUTE block:

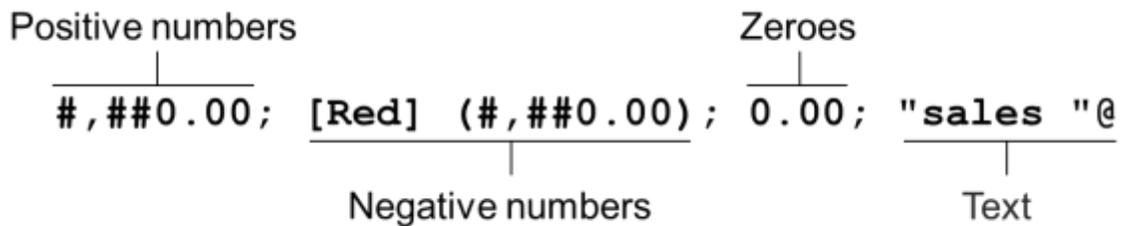
```
compute change / character length=1;
  if (visit5_lbstresn gt baseline_lbstresn)
    then change = 'E1'x;
  else if (visit5_lbstresn lt baseline_lbstresn)
    then change = 'E2'x;
endcomp;
```

With all of the code modifications in place, the resulting workbook matches the output shown in [Figure 1](#).

**ExcelXP tagset users:** Refer to the "Using Excel Number Formats" section, below, for information about applying formats to numeric values.

### USING EXCEL NUMBER FORMATS

The Excel ODS destination automatically applies Excel number formats to numeric columns based on the data and formats specified in the procedure code. Figure 10 shows the general structure of Excel number formats (Microsoft Corporation 2016b). The number sign (#) in an Excel format represents a numeric digit, excluding insignificant zeros. A zero (0) displays a numeric digit, including insignificant zeros. Use zeros in Excel formats when you want to retain leading or trailing zeros.



**Figure 10. Structure of Excel Number Formats**

Table 4 lists the data types and formats automatically applied by the Excel destination to create the output in [Figure 1](#).

Column Label	SAS Data Type	SAS Format	Excel Format
Subject Identifier	Character		General
Age	Numeric		#####0
Baseline Result	Numeric	5.2	#0.00
Unit	Character		General
Visit 5 Result	Numeric	5.2	#0.00
Unit	Character		General
Change from Baseline	Character		General
Baseline Date/Time of Specimen Collection	Character		General
Visit 5 Date/Time of Specimen Collection	Character		General

**Table 4. Data Types and Formats for Columns in Figure 1**

The Excel format that is equivalent to the SAS 5.2 format specified in the DEFINE statements for the **Baseline Result** and **Visit 5 Result** columns is #0.00. For example, the female baseline result value 6.10296 is displayed as 6.10 in [Figure 1](#).

Because no format is associated with the AGE column, PROC REPORT applies the BESTw. SAS format using the default value of the COLWIDTH option (9) for w. The #####0 Excel format is equivalent to the BEST9. SAS format (SAS Institute Inc. 2015a, 2015d).

The ExcelXP tagset does not automatically apply Excel number formats, but you can specify Excel formats using the TAGATTR style attribute in an ODS style override as follows:

```
define baseline_lbstresn / display 'Baseline*Result' format=5.2
                        style(column)=[tagattr='format:#0.00'];

define visit5_lbstresn / display 'Visit 5*Result' format=5.2
                        style(column)=[tagattr='format:#0.00'];
```

Be sure to quote the entire attribute value and include the **format:** keyword. You can also use the TAGATTR attribute with the Excel destination.

## THE FINAL SAS CODE

The final SAS code to create the output of [Figure 1](#) follows, with changes from the initial SAS code highlighted:

```
ods _all_ close;

ods Excel file='LabResults.xlsx' style=HTMLBlue;

* Option that affects all worksheets;

ods Excel options(embedded_titles='yes');

* First worksheet;

ods Excel options(sheet_name='Female - Plot');

title1 'Change in Cholesterol from Baseline for Females - Xanomeline High
       Dose';
title2 'Shaded area indicates normal range';

proc sgplot data=sample.Baseline_Visit5
            sganno=sample.F_Cholesterol_Anno
            noautolegend;
  where (sex eq 'F' and lbtest eq 'CHOLESTEROL');
  scatter x=visit5_lbstresn y=baseline_lbstresn;
  xaxis values=(4 to 8 by 1) minor minorcount=4;
  yaxis values=(4 to 8 by 1) minor minorcount=4;
  lineparm x=4 y=4 slope=1;
  label visit5_lbstresn = 'Cholesterol at Visit 5 (mmol/L)'
        baseline_lbstresn = 'Cholesterol Baseline (mmol/L)';
run; quit;

* Second worksheet;

ods Excel options(sheet_name='Female - Report');

title2;

proc report data=sample.Baseline_Visit5 nowd split='*'
           style(column)=[just=center];
  where (sex eq 'F' and lbtest eq 'CHOLESTEROL');
  column usubjid age baseline_lbstresn baseline_lbstresu
         visit5_lbstresn visit5_lbstresu change
         baseline_lbdtc visit5_lbdtc;
```

```

define usubjid          / order      'Subject*Identifier';
define age              / display    '*Age';
define baseline_lbstresn / display  'Baseline*Result' format=5.2;
define baseline_lbstresu / display  'Unit';
define visit5_lbstresn  / display  'Visit 5*Result' format=5.2;
define visit5_lbstresu  / display  'Unit';
define change           / computed  'Change from*Baseline'
                        style(column)=[fontfamily=Wingdings];
define baseline_lbdtc   / display  'Baseline Date/Time of*Specimen
Collection';
define visit5_lbdtc     / display  'Visit 5 Date/Time of*Specimen
Collection';

compute change / character length=1;
  if (visit5_lbstresn gt baseline_lbstresn)
    then change = 'E1'x;
  else if (visit5_lbstresn lt baseline_lbstresn)
    then change = 'E2'x;
endcomp;

run; quit;

* Third worksheet;

ods Excel options(sheet_name='Male - Plot');

title 'Change in Cholesterol from Baseline for Males - Xanomeline High
Dose';
title2 'Shaded area indicates normal range';

proc sgplot data=sample.Baseline_Visit5
            sganno=sample.M_Cholesterol_Anno
            noautolegend;
  where (sex eq 'M' and lbtest eq 'CHOLESTEROL');
  scatter x=visit5_lbstresn y=baseline_lbstresn;
  xaxis values=(3 to 9 by 1) minorcount=4;
  yaxis values=(3 to 9 by 1) minorcount=4;
  lineparm x=3 y=3 slope=1;
  label visit5_lbstresn   = 'Cholesterol at Visit 5 (mmol/L)'
        baseline_lbstresn = 'Cholesterol Baseline (mmol/L)';
run; quit;

* Fourth worksheet;

ods Excel options(sheet_name='Male - Report');

title2;

proc report data=sample.Baseline_Visit5 nowd split=''
            style(column)=[just=center];
  where (sex eq 'M' and lbtest eq 'CHOLESTEROL');
  column usubjid age baseline_lbstresn baseline_lbstresu
         visit5_lbstresn visit5_lbstresu change
         baseline_lbdtc visit5_lbdtc;
  define usubjid          / order      'Subject*Identifier';
  define age              / display    '*Age';
  define baseline_lbstresn / display  'Baseline*Result' format=5.2;

```

```

define baseline_lbstresu / display 'Unit';
define visit5_lbstresn / display 'Visit 5*Result' format=5.2;
define visit5_lbstresu / display 'Unit';
define change / computed 'Change from*Baseline'
                    style(column)=[fontfamily=Wingdings];
define baseline_lbdtc / display 'Baseline Date/Time of*Specimen
                               Collection';
define visit5_lbdtc / display 'Visit 5 Date/Time of*Specimen
                              Collection';

compute change / character length=1;
    if (visit5_lbstresn gt baseline_lbstresn)
        then change = 'E1'x;
    else if (visit5_lbstresn lt baseline_lbstresn)
        then change = 'E2'x;
endcomp;

run; quit;

ods Excel close;

```

## SAS SERVER TECHNOLOGY

You can deliver dynamically generated SAS output into Excel using the Application Dispatcher or the SAS® Stored Process Server. The Application Dispatcher is part of SAS/IntrNet® software. The SAS Stored Process Server is available starting with SAS®9 as 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).

These products enable you to execute SAS programs from a Web browser or any other client that can open an HTTP connection to the Application Dispatcher or the SAS Stored Process Server. Both of these products 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, macro, or SCL code. Thus, all of the code that has been shown up to this point can be executed by both the Application Dispatcher and the SAS Stored Process 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 11](#) shows a Web page that can deliver SAS output directly to Excel, using a Web browser as the client.

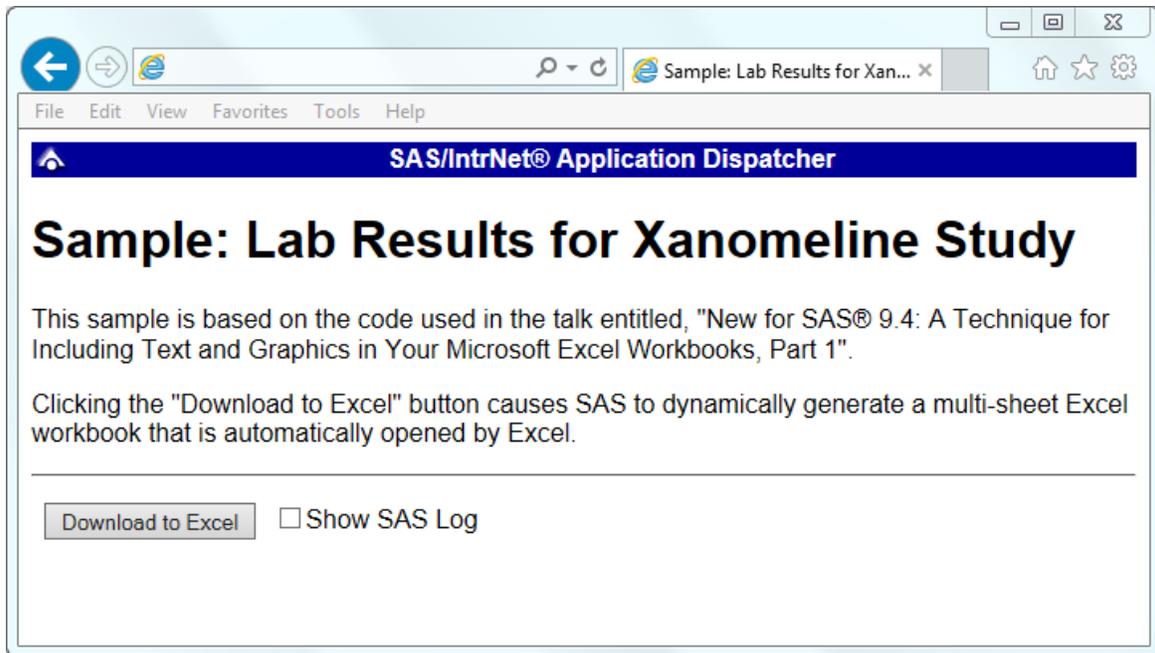


Figure 11. Web Page to Drive a SAS/IntrNet Application

Clicking **Download to Excel** executes a slightly modified version of the SAS code that we have been working on. The modifications are as follows:

```
%let RV=%sysfunc(appsrv_header(Content-type,application/vnd.ms-excel));
%let RV=%sysfunc(appsrv_header(Content-disposition,attachment; filename=
"LabResults.xlsx")); * Ignore line wrapping;

ods _all_ close;

ods Excel file=_webout style=HTMLBlue;

* Remainder of the "final" SAS code;

ods Excel close;
```

The first APPSRV\_HEADER function sets a MIME header that causes the SAS output to be opened by Excel, instead of being rendered by the Web browser. This statement is required.

The second APPSRV\_HEADER function sets a MIME header that causes the filename to be displayed in the File Download dialog box. As you can see in Figure 12, the filename appears as **LabResults.xlsx**. This header might cause problems with some versions of Excel, so be sure to test your applications before deploying them in a production environment. This statement is optional.



Figure 12. File Download Dialog Box

The reserved `_WEBOUT` fileref is automatically defined by the SAS server and is always used to direct 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.

**ExcelXP tagset users:** Be sure to use the **tagsets.ExcelXP** destination and **.xml** for the file extension. No other code changes are needed.

When you click the **Download to Excel** button on the Web page you are presented with the File Download dialog box (Figure 12). You can then click **Open** to immediately open your SAS output using Excel, or you can click **Save**.

This paper presents one example of how you can dynamically deliver SAS output to Excel. For more detailed information and other examples, see the SAS/IntrNet Application Dispatcher and SAS Stored Process documentation (SAS Institute Inc. 2013, 2015e), as well as this author's earlier papers (DelGobbo 2002, 2003).

## CONCLUSION

The SAS Excel ODS destination available in the third maintenance for SAS 9.4 and later provides an easy way to export your SAS data and graphic images to Excel workbooks in the XLSX format. By using ODS options and style overrides you can customize the output to achieve your design goals. Similar results are achieved in earlier versions of SAS using the ExcelXP tagset.

## REFERENCES

- DelGobbo, Vincent. 2002. "Techniques for SAS<sup>®</sup> Enabling Microsoft Office in a Cross-Platform Environment". *Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference*. Cary, NC: SAS Institute Inc. Available at <http://www2.sas.com/proceedings/sugi27/p174-27.pdf>.
- DelGobbo, Vincent. 2003. "A Beginner's Guide to Incorporating SAS<sup>®</sup> Output in Microsoft Office Applications". *Proceedings of the Twenty-Eighth Annual SAS Users Group International Conference*. Cary, NC: SAS Institute Inc. Available at <http://www2.sas.com/proceedings/sugi28/052-28.pdf>.
- DelGobbo, Vincent. 2008. "Tips and Tricks for Creating Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS<sup>®</sup>". *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. Available at <http://www2.sas.com/proceedings/forum2008/192-2008.pdf>.
- DelGobbo, Vincent. 2009. "More Tips and Tricks for Creating Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS<sup>®</sup>". *Proceedings of the SAS Global Forum 2009 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings09/152-2009.pdf>.
- DelGobbo, Vincent. 2010. "Traffic Lighting Your Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS<sup>®</sup>". *Proceedings of the SAS Global Forum 2010 Conference*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/resources/papers/proceedings10/153-2010.pdf>.
- DelGobbo, Vincent. 2011. "Creating Stylish Multi-Sheet Microsoft Excel Workbooks the Easy Way with SAS<sup>®</sup>". *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. 2015. "Vince DelGobbo's ExcelXP Tagset Paper Index". Available at <http://support.sas.com/community/events/sastalks/presentations/ExcelXPPaperIndex.pdf>
- Microsoft Corporation. 2001. "XML Spreadsheet Reference". Available at [http://msdn.microsoft.com/en-us/library/aa140066\(office.10\).aspx](http://msdn.microsoft.com/en-us/library/aa140066(office.10).aspx).
- Microsoft Corporation. 2016a. "Error opening file: 'The file format differs from the format that the file name extension specifies'". Available at <http://support.microsoft.com/kb/948615>
- Microsoft Corporation. 2016b. "Number format codes". Available at <https://support.office.com/en-us/article/Number-format-codes-5026bbd6-04bc-48cd-bf33-80f18b4eae68>.
- SAS Institute Inc. 2008. "SAS<sup>®</sup> 9 Reporting Procedure Styles Tip Sheet". Available at <http://support.sas.com/rnd/base/ods/scratch/reporting-styles-tips.pdf>.
- SAS Institute Inc. 2009. "Sample 36900: Instructions for viewing all of the style templates that are shipped with SAS<sup>®</sup>". Available at <http://support.sas.com/techsup/notes/v8/36/900.html>

SAS Institute Inc. 2013. *SAS/IntrNet® 9.4: Application Dispatcher*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/dispatch/64895/HTML/default/viewer.htm#p06h82ux8glu1pn16k9dxw8tjpyz.htm>.

SAS Institute Inc. 2015a. "DEFINE Statement". *Base SAS® 9.4 Procedures Guide, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at <https://support.sas.com/documentation/cdl/en/proc/68954/HTML/default/viewer.htm#p0wy1vqwvz43uhn1g77eb5xlvzqh.htm>

SAS Institute Inc. 2015b. "FILESYSYEM= System Option: z/OS". *SAS® 9.4 Companion for z/OS, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/hosto390/68955/HTML/default/viewer.htm#n0b7vr7xilcwccn13bv8r2fk9yco.htm>

SAS Institute Inc. 2015c. "ODS EXCEL Statement". *SAS® 9.4 Output Delivery System: User's Guide, Fourth Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/odsug/67921/HTML/default/viewer.htm#p09n5pw9ol0897n1qe04zeur27rv.htm>

SAS Institute Inc. 2015d. "PROC REPORT Statement". *Base SAS® 9.4 Procedures Guide, Fifth Edition*. Cary, NC: SAS Institute Inc. Available at <https://support.sas.com/documentation/cdl/en/proc/68954/HTML/default/viewer.htm#n1dz7jdasx5t56n1rm1x346dyk6n.htm>

SAS Institute Inc. 2015e. *SAS® 9.4 Stored Processes: Developer's Guide, Third Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/stpug/67499/HTML/default/viewer.htm#n180km1hadyuton13gx4bzlqhpwr.htm>.

SAS Institute Inc. 2015f. "Style Attributes Detailed Information". *SAS® 9.4 Output Delivery System: User's Guide, Fourth Edition*. Cary, NC: SAS Institute Inc. Available at <http://support.sas.com/documentation/cdl/en/odsug/67325/HTML/default/viewer.htm#p0xi2cygmfk0wkn1ei625zq5r488.htm>.

## ACKNOWLEDGMENTS

The author would like to thank Chris Barrett of SAS Institute for his valuable contributions to this paper.

## CONTACT INFORMATION

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

Vincent DelGobbo  
SAS Institute Inc.  
100 SAS Campus Drive  
Cary, NC 27513



[sasvcd@SAS.com](mailto:sasvcd@SAS.com)

<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 (from [support.sas.com/sasusersupport/usergroups/support](http://support.sas.com/sasusersupport/usergroups/support)) at least eight weeks in advance.

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.