



## Creating Multi-Sheet Microsoft Excel Workbooks with SAS®: The Basics and Beyond, Part 1

---

Vince DelGobbo, SAS



**SAS Global Forum 2014**

**March 23-26, 2014**

**Gaylord National Resort and Conference Center**

**Washington, DC**

A special “thank you” to Marje Fecht for inviting me to present this topic, and to Chris Barrett of SAS Institute Inc. for his valuable contributions to the accompanying paper.

## Goals

- Integrate SAS output w/ Excel
- Give you something you can use TODAY



## Agenda

- Review sample data
- ODS basics
- Generating XML for Excel
- Open output in Excel
- Fix formatting issues
- Using SAS server technology



## Software Requirements

- Base SAS, *any* operating system
- SAS 9.1.3 Service Pack 4 or later
- Microsoft Excel 2002 or later (a.k.a. Excel XP)



# SAS Output Viewed with Excel

The screenshot shows an Excel spreadsheet titled "LabResults.xml - Microsoft Excel". The active worksheet is "WEEK 1" and the data is for "Subject 01-701-1015". The spreadsheet contains the following data:

Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
1	LAB BASELINE	12/26/2013 2:45 PM	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5

The second worksheet, "WEEK 2", shows a single row of data for Visit 4, Visit Name WEEK 2, Collection Date/Time 1/16/2014 1:17 PM, Test ALB, Result 3.9, Units g/dl, and Range 3.3 - 4.9.

The workbook includes four worksheets containing fictional lab results data over time for a single patient. A SAS title replaces standard BY line text, and an Excel format, not a SAS format, is used to control the appearance of the datetime values in the "Collection Date/Time" column. SAS datetime values are automatically converted to Excel datetime values.

## Representative Values in Sample Data

Column Name	Column Label	Column Type	Typical Values
LBCAT	Category for Lab Test	Character	CHEMISTRY, HEMATOLOGY, OTHER, URINALYSIS
VISITNUM	Visit	Numeric	1, 4, 8, 10, 13
VISIT	Visit*Name	Character	LAB BASELINE, WEEK 2, WEEK 4, WEEK 26
VISIT_DATETIME	Collection* Date/Time	Numeric	1703688300, 1707828960, 1719920700
LBTESTCD	Test	Character	BILI, CHOL, EOS, WBC
LBORRES	Result	Character	0.05, 0.9, 1.013, 8.8, 15, 140
LBORRESU	Units	Character	%, MILL/uL, NO UNITS, g/dL, pg/ml
RANGE	Range	Character	1.006 - 1.03, 21 - 169, 156 - 300, not available
LBNRIND	<blank>	Character	A, H, L, N

6

Here is an abbreviated list of column properties for the LabResults SAS table that is used to create the Excel workbook.

An asterisk (\*) is used as a split character in some variable labels to control text wrapping in the column headings, and the values in the VISIT\_DATETIME column are SAS datetime values.

## General Steps

1. Run SAS code to create output
2. Store output where Excel can access it
3. Open output with Excel
4. Modify SAS code to correct formatting problems



We use ODS to create an XML file that is stored in a location that Excel can access. In your production system, SAS and Excel may reside on two different systems. Thus, you may have to make use of network drives, FTP, or some other means to move the SAS output to a location that Excel can access.

Then the ODS output is opened using Excel. If you have ever done this before, you have probably encountered formatting problems. We fix those problems, and then explore techniques to instruct ODS to create output that is more attractive.

## ODS Basics

- Part of Base SAS
- Easily generate multiple output types (HTML, RTF, PDF, XML, etc.)
- A "destination" creates the actual output
- A "style" controls the appearance
- Usage:

HTML or RTF or PDF ...

```
ods DestName style=StyleName file=... ;  
  * Your SAS procedure code here;  
ods DestName close;
```



8

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, etc.). An ODS *style* controls the appearance of the output (colors, fonts, border lines, etc.).

Both a destination and a style are needed to generate output. If you do not specify a style, a default style is used.

We use a special type of ODS destination called a "tagset". ODS tagsets can be modified to meet your specific needs using the `TEMPLATE` procedure, which is part of the Base SAS product. And you can use the `TEMPLATE` procedure to create your own tagsets.

### TEMPLATE Procedure Overview

**SAS 9.3:** <http://tinyurl.com/mnzeok3>

**SAS 9.4:** <http://tinyurl.com/pz6zqr4>

## ODS Basics – Output for Excel

- Excel can open specially made XML files as multi-sheet workbooks (graphics not supported)

- Use the ExcelXP tagset and Printer style:

```
ods listing close;  
ods tagsets.ExcelXP style=Printer file=... ;  
* Your SAS procedure code here;  
ods tagsets.ExcelXP close;
```



9

We use the ODS tagset named ExcelXP to create XML output that can be opened using Microsoft Excel 2002 and later. When opened with Excel, the XML file is rendered as a multi-sheet Excel workbook. Additionally, we use an ODS style named Printer, supplied by SAS, to control the appearance of the output.

Because the ExcelXP ODS tagset creates files that conform to the Microsoft XML Spreadsheet Specification, you can create multi-sheet Excel workbooks containing the output from almost any SAS procedure. The exception is that the Microsoft XML Spreadsheet Specification does not support images, so the output from graphics procedures cannot be used.

### Microsoft XML Spreadsheet Reference

<http://tinyurl.com/3p6sked>

## Basic SAS Code

```
title 'Lab Results for Subject 01-701-1015';

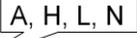
proc report data=sample.LabResults nowd split='*';
  by lbcats;

  column ... ;
  define ... ;

  compute range / char length=13;
    * Create Low-High range (Ex. 3.3 - 4.9);
  endcomp;

  compute lbnrind;
    if (upcase(lbnrind) eq 'N') then lbnrind = '';
  endcomp;

  compute after visitnum;
    line '';
  endcomp;
run; quit;
```



10

The COLUMN statement specifies the order to display the columns, and the column roles, justification, labels, and formats are specified in the DEFINE statements.

The first COMPUTE block constructs the range if both low and high values are available for the test, otherwise "not available" is used for the value.

The second block suppresses the display of the LBNRIND value for tests with normal results, and the third block inserts a blank line between office visits.

A complete listing of the SAS code is in the accompanying paper.

## Using ODS and the ExcelXP Tagset

```
ods tagsets.ExcelXP file='LabResults.xml' style=Printer;  
  
title 'Lab Results for Subject 01-701-1015';  
  
proc report data=sample.LabResults nowd split='*';  
  by lbcats;  
  
  column ... ;  
  
  define ... ;  
  
  compute ... ;  
  
run; quit;  
  
ods tagsets.ExcelXP close;
```



11

The ExcelXP tagset generates the output, and the PRINTER style controls the appearance of the output.

By default, the ExcelXP tagset creates a new worksheet when a SAS procedure creates new tabular output. PROC REPORT is run with a BY statement and creates four tables, one for each distinct value of the LBCAT variable. Each table is created in a separate worksheet.

The last ODS statement closes the ExcelXP destination and releases the XML file so that it can be opened with Excel.

## Open LabResults.xml with Excel

- Open Excel: **Start > Programs > ...**
- **File > Open**
- Navigate to ...\**LabResults.xml** and click **Open**

~ OR ~

- Navigate to output directory and double-click **LabResults.xml**



12

The XML file created by ODS can now be opened with Microsoft Excel.

Excel reads and converts the XML file to the Excel format. After the conversion, you can perform any Excel function on the data.

To save a copy of the file in Excel binary (xls) format using Excel 2002, 2003 or 2010, select **File → Save As** and then, from the **Save as type** drop-down list, select **Microsoft Excel Workbook (\*.xls)**. If you're using Excel 2007, click the Microsoft Office Button, and then select **Save As → Excel 97-2003 Workbook**.

If you're using Excel 2007 or 2010 and want to save the document in the Microsoft Office Open XML format, choose **Excel Workbook (\*.xlsx)** from the **Save as type** drop-down list.

## LabResults.xml Viewed with Excel

The screenshot shows an Excel spreadsheet with the following data:

Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
1	LAB BASELINE	2013-12-26T14:45:00	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5
4	WEEK 2	2014-01-16T13:17:00	ALB	3.9	g/dl	3.3 - 4.9
			ALP	50	U/L	35 - 115

Here are the results of executing the basic SAS code, and then opening the resulting LabResults.xml file with Excel. This output resembles the desired format, but has the following problems:

1. Unattractive, default worksheet names are used.
2. Title text is missing.
3. Standard BY line text ("Category for Lab Test=CHEMISTRY") precedes the table.
4. The collection date and time values are displayed differently.

We can now change the basic SAS code to correct these problems.

## Run Setup.sas

1. Start SAS
2. **File > Open Program**
3. Select **Setup.sas** and then click **Open**
4. Review code and then submit



14

The SAMPDIR global macro variable specifies the directory containing our sample code and data, as well as the ODS-generated XML output.

The program assigns a SAS library (SAMPLE) for the input data, using the SAMPDIR macro variable.

## Ex. 1 – Creating the Initial Workbook

1. Go to SAS
2. **File > Open Program > Exercise1.sas**
3. Review code and then submit
4. Start Excel, and then **File > Open**
5. Navigate to **C:\HOW\DelGobbo\**
6. Double click **LabResults.xml**
7. Close document; keep Excel open



### TO DO:

Review and submit the code, and then view the resulting XML file using Excel.

## Understanding and Using the ExcelXP Tagset Options



16

The following techniques apply to all SAS procedure output.

# Change the Worksheet Names

The screenshot shows a Microsoft Excel window titled 'LabResults.xml - Microsoft Excel'. The active worksheet is 'Category for Lab Test=CHEMISTRY'. The data is organized into a table with the following columns: Visit, Visit Name, Collection Date/Time, Test, Result, Units, and Range. The table contains 20 rows of data for various lab tests. A red arrow points to the worksheet tab 'Table 2 - Category for Lab Test' at the bottom of the window.

Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
1	LAB BASELINE	2013-12-26T14:45:00	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5
4	WEEK 2	2013-12-16T13:17:00	ALB	3.9	g/dl	3.3 - 4.9
			ALP	50	U/L	35 - 115

ODS generates a unique name for each worksheet, as required by Excel.

However, the names are generally not very attractive. There are several tagset options that you can use to alter the names of the worksheets.

## ExcelXP Tagset Options

- Syntax: `options(option-name='option-value')`
- Example: Hard-coding worksheet name

```
options(sheet_name='worksheet-name');
```

- Option values must be quoted
- Can have multiple ODS statements
- Options remain in effect until changed



18

The ExcelXP tagset supports many options that control both the appearance and functionality of the Excel workbook. Many of these tagset options are simply tied directly to existing Excel options or features. Tagset options are specified in an ODS statement using the `OPTIONS` keyword, as shown above.

For example, ODS automatically generates unique worksheet names, but we can use the `SHEET_NAME` option to explicitly specify a worksheet name. In our case we don't want to hard code the names, so a different set of options are used.

**IMPORTANT NOTE:** Tagset options remain in effect until they are set to another value, or until the ODS destination is closed.

## Change the Worksheet Names

- **SHEET\_INTERVAL** – When to create worksheets
- **SHEET\_LABEL** – Prefix for worksheet name

CHEMISTRY, HEMATOLOGY, OTHER, URINALYSIS

```
ods tagsets.ExcelXP options(sheet_interval='bygroup'  
                             sheet_label=' ');
```



19

ODS generates a unique name for each worksheet, as required by Excel. There are, however, several tagset options that you can use to alter the names of the worksheets.

The SHEET\_INTERVAL option controls the interval at which SAS output is placed into worksheets, and SHEET\_LABEL is used to specify the prefix for the worksheet names.

When used together as shown here, the current value of the first BY variable, without any prefix, is used in the worksheet name.

## Change the Worksheet Names

```
ods tagsets.ExcelXP file='LabResults.xml' style=Printer;  
  
title 'Lab Results for Subject 01-701-1015';  
  
ods tagsets.ExcelXP options(sheet_interval='bygroup'  
                             sheet_label=' ');  
  
proc report data=sample.LabResults nowd split='*';  
  by lbcatt;  
  
  column ... ;  
  define ... ;  
  compute ... ;  
  
run; quit;  
  
ods tagsets.ExcelXP close;
```



20

The additional ODS statement to handle the tagset options.

NOTE: When specifying *additional* ODS statements, do not specify the FILE, STYLE, or any other keyword or option that is supported by ODS. Those options should be specified only in the initial ODS statement.

## Change the Worksheet Names



The worksheet names now match the desired output.

## Ex. 2 – Change the Worksheet Names

1. Go to SAS
2. **File > Open Program > Exercise2.sas**
3. Follow TO DO instructions and then submit
4. Go to Excel
5. **File > Recent**, and then select **LabResults.xml**
6. Note worksheet names
7. Close document; keep Excel open



### TO DO:

Line 13: Specify the value for the SHEET\_INTERVAL option to create a separate worksheet for each unique value of the BY variable LBCAT.

## Display Title in Worksheet

- Title text → Excel print header
- Can control location of title text:

```
embedded_titles='yes'
```



23

By default, SAS titles and footnotes appear as Excel print headers and print footers, respectively, which are displayed when the Excel document is printed. You can confirm this by viewing the Excel **Header/Footer** tab of the Page Setup dialog box.

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

## Display Title in Worksheet

```
ods tagsets.ExcelXP file='LabResults.xml' style=Printer;  
  
title 'Lab Results for Subject 01-701-1015';  
  
ods tagsets.ExcelXP options(sheet_interval='bygroup'  
                           sheet_label=' '  
                           embedded_titles='yes');  
  
proc report data=sample.LabResults nowd split='*';  
  by lbcats;  
  
  column ... ;  
  define ... ;  
  compute ... ;  
  
run; quit;  
  
ods tagsets.ExcelXP close;
```



# Display Title in Worksheet

LabResults.xml - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Developer SAS

C5 2013-12-26T14:45:00

1 **Lab Results for Subject 01-701-1015**

2

3 **Category for Lab Test=CHEMISTRY**

Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
1	LAB BASELINE	2013-12-26T14:45:00	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5
4	WEEK 2	2014-01-16T13:17:00	ALB	3.9	g/dl	3.3 - 4.9

Ready

25

BY line text appears in the worksheets because the REPORT procedure is executed with a BY statement. However, this text is redundant because the BY value is displayed in the worksheet name.

## Suppress BY Line Text

- Do not use NOBYLINE *system* option
- Use SUPPRESS\_BYLINES tagset option:

```
suppress_bylines='yes'
```



26

To omit the BY line text, specify the SUPPRESS\_BYLINES option in the ODS statement.

Do not use the NOBYLINE *system* option, because this disables BY group processing in the ExcelXP tagset, even though the SHEET\_INTERVAL tagset option is set to "bygroup".

## Suppress BY Line Text

```
ods tagsets.ExcelXP file='LabResults.xml' style=Printer;

title 'Lab Results for Subject 01-701-1015';

ods tagsets.ExcelXP options(sheet_interval='bygroup'
                           sheet_label=' '
                           embedded_titles='yes'
                           suppress_bylines='yes');

proc report data=sample.LabResults nowd split='*';
  by lbcatt;

  column ... ;
  define ... ;
  compute ... ;

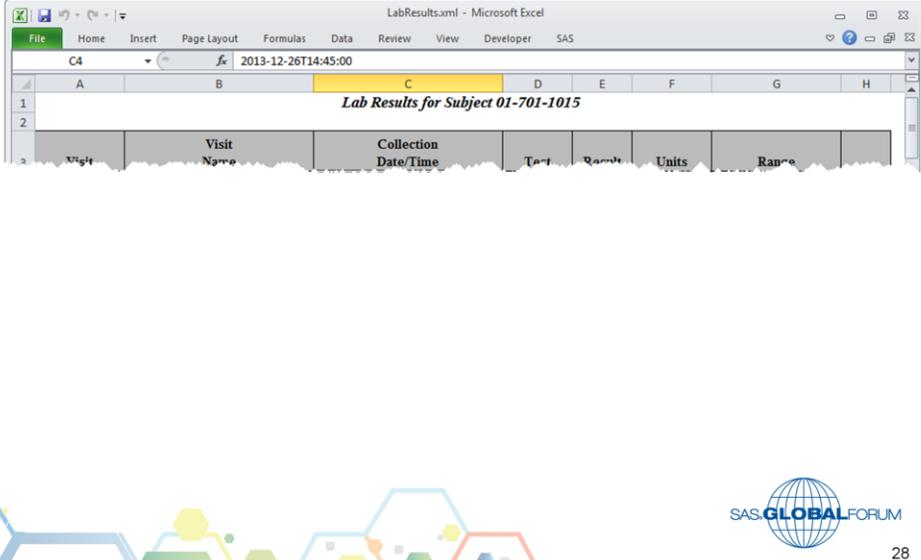
run; quit;

ods tagsets.ExcelXP close;
```



27

## Suppress BY Line & Display Title



The screenshot shows a Microsoft Excel spreadsheet titled "LabResults.xml - Microsoft Excel". The active cell is C4, containing the date and time "2013-12-26T14:45:00". The spreadsheet displays a table with the following structure:

<i>Lab Results for Subject 01-701-1015</i>							
Visit	Visit Name	Collection Date/Time	Test	Results	Units	Range	

The table is partially visible, with the first two rows shown. The first row is the title row, and the second row is the header row. The table is followed by a decorative footer graphic and the SAS GLOBAL FORUM logo.

Only the title text precedes the table.

## Ex. 3 – Suppress BY Line & Display Title

1. Go to SAS
2. **File > Open Program > Exercise3.sas**
3. Follow TO DO instructions and then submit
4. Go to Excel
5. **File > Recent**, and then select **LabResults.xml**
6. Note title in worksheets
7. Close document; keep Excel open



### TO DO:

Line 15: Specify the value for the EMBEDDED\_TITLES option to display title text within the worksheets.

Line 16: Specify the option to suppress BY line text.

# Understanding and Using ODS Style Overrides



The techniques described *in this section only* work only with the PRINT, REPORT, and TABULATE procedures.

## SAS Datetime → Excel Datetime, Pt. 1

### Collection Date/Time

2014-01-16T13:17:00

- SAS & Excel use different date/time systems ✓
- Excel prefers ISO 8601 datetime (SAS E8601DT format)
- But ... ODS specifies **String** for data type in XML

```
<Cell ss:StyleID="data__1" ss:Index="3">  
  <Data ss:Type="String">2014-01-16T13:17:00</Data>  
</Cell>
```

Needs to be **DateTime**



31

SAS and Microsoft Excel use different date and time systems. Consequently, you often encounter problems when Excel reads SAS output containing datetime values.

One way to correct this behavior is to write SAS code to convert numeric SAS datetime values to numeric Excel datetime values, but this approach is problematic because you must alter your original SAS data.

A better solution is to allow automatic datetime conversion to take place between SAS and Excel.

Because Excel expects datetime values to be represented using the ISO 8601 format (yyyy-mm-ddThh:mm:ss), SAS datetime values should be formatted using the IS8601DT format. However, open the LabResults.xml file using a text editor and you will see that the datetime cells are specified to be **String** type. You cannot apply Excel formats or perform mathematical operations on the data in these cells.

We use an ODS style override to fix this problem.

## ODS Style Overrides

- Supported by PRINT, REPORT, and TABULATE
- Change any ODS style attribute via STYLE=
- Example:

```
define visit /  
  style(column) = [font_style = italic  
                  background = orange]
```

Location                      Attributes

- Refer to the ODS documentation for a list of supported attributes
- Refer to PRINT, REPORT, and TABULATE doc for sample usage



Style overrides are supported by the PRINT, REPORT, and TABULATE procedures, and can be specified in several ways, with the most common shown here.

This example overrides the font style and the cell background color attributes for the data values in the VISIT column.

# ODS Style Overrides – Example Output

LabResults.xml - Microsoft Excel

2013-12-26T14:45:00

*Lab Results for Subject 01-701-1015*

Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
1	LAB BASELINE	2013-12-26T14:45:00	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5
4	WEEK 2	2014-01-16T13:17:00	ALB	3.9	g/dl	3.3 - 4.9

CHEMISTRY / HEMATOLOGY / OTHER / URINALYSIS

Ready 100%

33

## Style Attributes

**SAS 9.3:** <http://tinyurl.com/nufsw4o>

**SAS 9.4:** <http://tinyurl.com/kj5zqy3>

## PRINT Procedure Examples

**SAS 9.3:** <http://tinyurl.com/ct3j7t4>

**SAS 9.4:** <http://tinyurl.com/l74t9rr>

## REPORT Procedure Examples

**SAS 9.3:** <http://tinyurl.com/axnqrd4>

**SAS 9.4:** <http://tinyurl.com/mv8hzuu>

## TABULATE Procedure Examples

**SAS 9.3:** <http://tinyurl.com/bdqgl4c>

**SAS 9.4:** <http://tinyurl.com/o4ojttr>

## Location Values for PROC REPORT

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		

34

Style overrides are applied to specific parts of your SAS output. These specific parts of your SAS output are called *locations*.

The COLUMN location of PROC REPORT controls the appearance of data cells.

### SAS® 9 Reporting Procedure Styles Tip Sheet

<http://tinyurl.com/3cfqdv9>

## SAS Datetime → Excel Datetime, Pt. 1

- Current DEFINE statement:

```
define visit_datetime / order left 'Collection*Date/Time'  
  format=e8601dt.; ✓
```

- Apply style override to specify data type:

```
define visit_datetime / order left 'Collection*Date/Time'  
  format=e8601dt.  
  style(column)=[tagattr='type:DateTime'];
```



35

Because Excel does not recognize the value as a datetime value, you cannot apply Excel formats or perform mathematical operations on the data.

To fix this problem, use the TAGATTR attribute in a style override to specify that the value is an Excel **DateTime** type.

**NOTE:** The value specified in the style override, **DateTime**, is case-sensitive.

## SAS Datetime → Excel Datetime, Pt. 1

```
ods tagsets.ExcelXP file='LabResults.xml' style=Printer;  
  
title 'Lab Results for Subject 01-701-1015';  
  
ods tagsets.ExcelXP options(...);  
  
proc report data=sample.LabResults nowd split='*';  
  by lbcats;  
  
  column ... ;  
  
  define visit_datetime / order left 'Collection*Date/Time'  
    format=e8601dt.  
    style(column)=[tagattr='type:DateTime'];  
  
  compute ... ;  
  
run; quit;  
  
ods tagsets.ExcelXP close;
```



36

## SAS Datetime → Excel Datetime, Pt. 1

LabResults.xml - Microsoft Excel

41634.614583333

Lab Results for Subject 01-701-1015

Visit	Visit Name	Collection Date	Test	Result	Units	Range
1	LAB BASELINE	41634.61438	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5
4	WEEK 2	41655.55347	ALB	3.9	g/dl	3.3 - 4.9

37

After the automatic conversion, the cells now contain unformatted Excel datetime values.

## Ex. 4 – SAS Datetime → Excel Datetime

1. Go to SAS
2. **File > Open Program > Exercise4.sas**
3. Follow TO DO instructions and then submit
4. Go to Excel
5. **File > Recent**, and then select **LabResults.xml**
6. Note datetime value in cells; apply Excel format
7. Close document; keep Excel open



38

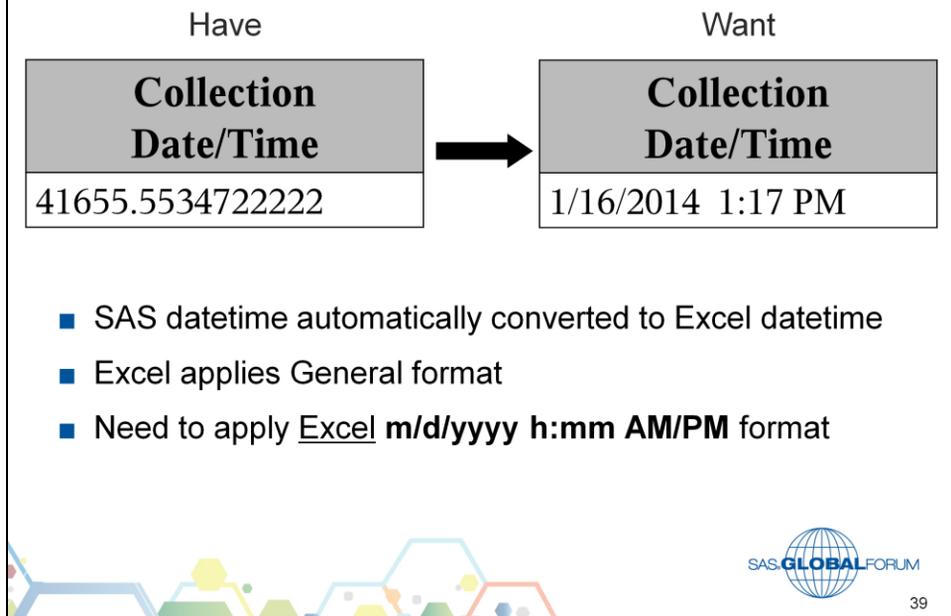
### TO DO:

Line 26: Specify the Excel data type **DateTime** for the value of the TAGATTR style attribute.

### Optional:

1. Click a cell containing a datetime value to select it.
2. Click the **Home** tab on the Excel ribbon.
3. Choose an Excel format from the **Number** group.

## SAS Datetime → Excel Datetime, Pt. 2



The SAS datetime value is automatically converted to an Excel datetime value, and Excel applies the "General" format.

We use an Excel format, specified in a style override, to control the display values in Excel.

## Some Excel Datetime Formats

Excel Format	Display Value
m/d/yyyy h:mm AM/PM *	1/16/2014 1:17 PM
mm/dd/yy hh:mm AM/PM	01/16/14 01:17 PM
ddmmyyy:hh:mm:ss	16Jan2014:13:17:00
mm/dd/yy	01/16/14
hh:mm AM/PM	01:17 PM
m/d/e	1/16/2014
yyyy-mm-ddThh:mm:ss	2014-01-16T13:17:00
mmm d, yyyy \@ h:mm AM/PM	Jan 16, 2014 @ 1:17 PM
mmmm d, yyyy \@ h:mm a/p	January 16, 2014 @ 1:17 p
ddd mmmmm d, yyyy	Thu J 16, 2014
dddd mmm d, yyyy	Thursday January 16, 2014
General *	41655.55347
#,###.0000	41,655.5535



This table shows how the Excel datetime value for January 16, 2014 at 1:17 pm is displayed in Excel using different formats.

Excel assigns the "General" format by default, and we use an ODS style override to specify the "m/d/yyyy h:mm AM/PM" format.

## SAS Datetime → Excel Datetime, Pt. 2

Apply style override to specify Excel format:

```
define visit_datetime / order left 'Collection*Date/Time'  
  format=e8601dt.  
  style(column)=[tagattr='type:DateTime  
                  format:m/d/yyyy h:mm AM/PM'];
```



41

## SAS Datetime → Excel Datetime, Pt. 2

LabResults.xml - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Developer SAS

C4 12/26/2013 2:45:00 PM

1  
2 *Lab Results for Subject 01-701-1015*

3	Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
4	1	LAB BASELINE	12/26/2013 2:45 PM	ALB	3.8	g/dl	3.3 - 4.9
5				ALP	34	U/L	35 - 115 L
6				ALT	27	U/L	6 - 34
7				AST	40	U/L	9 - 34 H
8				BILI	0.6	mg/dl	0.2 - 1.2
9				BUN	10	mg/dl	4 - 24
10				CA	8.8	mg/dl	8.4 - 10.3
11				CHOL	230	mg/dl	156 - 300
12				CK	70	U/L	21 - 169
13				CL	106	mEq/L	94 - 112
14				CREAT	0.9	mg/dl	0.7 - 1.4
15				GGT	15	U/L	5 - 50
16				GLUC	85	mg/dl	50 - 250
17				K	4.5	mEq/L	3.4 - 5.4
18				NA	140	mEq/L	135 - 145
19				PHOS	3.8	mg/dl	2.2 - 5.1
20				PROT	6.1	g/dl	6 - 8
21				URATE	4.5	mg/dl	2.5 - 7.5
22							
23	4	WEEK 2	1/16/2014 1:17 PM	ALB	3.9	g/dl	3.3 - 4.9

Ready

CHEMISTRY HEMATOLOGY OTHER URINALYSIS

100%

42

With all the code modifications in place, the resulting workbook matches the desired output.

## Ex. 5 – SAS Datetime → Excel Datetime

1. Go to SAS
2. **File > Open Program > Exercise5.sas**
3. Follow TO DO instructions and then submit
4. Go to Excel
5. **File > Recent**, and then select **LabResults.xml**
6. Note Excel-formatted datetime values
7. Close document; keep Excel open



43

### TO DO:

Line 26: Specify the Excel format **m/d/yyyy h:mm AM/PM** for the value of the TAGATTR style attribute.

## Open-Ended Exercise: Excel Formats

1. Go to SAS
2. Go to **Exercise5.sas**
3. Specify a different Excel format and then submit
4. Go to Excel
5. **File > Recent**, and then select **LabResults.xml**
6. Note Excel-formatted datetime values
7. Close document; keep Excel open



### TO DO:

Line 26: Specify various Excel formats, using the sample values shown on an earlier slide as a guide.

# Using SAS/IntrNet<sup>®</sup> and SAS Stored Processes



## SAS/IntrNet® and SAS Stored Processes

- SAS code is run from non-SAS client
- SAS is on any platform
- Client needs only a Web browser
- SAS output is delivered in real-time
- Web-enable the code we've been using



46

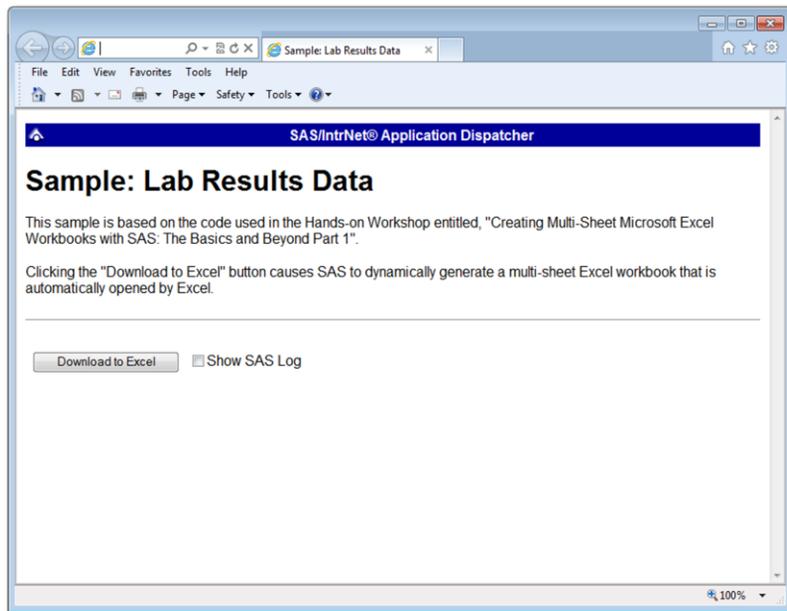
The purpose of the SAS/IntrNet® Application Dispatcher or SAS Stored Processes are to allow you to execute SAS programs from a client machine that does not have SAS installed. The client machine *may* have SAS installed, but that is not required.

A typical client-server model is followed. The SAS server can reside on any hardware platform (Windows, UNIX, z/OS, etc.) and is standing by, waiting to execute a SAS program. The most common client is a Web browser, again, running on any platform.

When the "OK" button of the Web page is clicked, input parameters, if any, are sent to the SAS server. Your SAS code executes, and the output is delivered in real-time to the Web browser.

The following slides illustrate this process, using a Web-enabled version of the SAS code we have been working on.

## Dynamically Generated XML



47

Here is a simple Web page that is used to execute SAS code stored on a server using the SAS/IntrNet Application Dispatcher.

The code that is executed is substantially similar to the final version of the code that we used to generate XML file, with a few changes to "Web-enable" it.

Refer to the accompanying paper for instructions on how to Web-enable your code.

Clicking **Download to Excel** executes the SAS program on the server.

## Dynamically Generated XML



48

Once the SAS program executes, the results are sent back to the Web browser.

Instead of the results being displayed in the Web browser, you are presented with a File Download dialog box.

You can click **Open** to immediately open your SAS output using Excel, or click **Save** to save a copy for later use.

# Dynamically Generated XML

LabResults.xml - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View Developer SAS

C4 12/26/2013 2:45:00 PM

1  
2 *Lab Results for Subject 01-701-1015*

Visit	Visit Name	Collection Date/Time	Test	Result	Units	Range
1	LAB BASELINE	12/26/2013 2:45 PM	ALB	3.8	g/dl	3.3 - 4.9
			ALP	34	U/L	35 - 115 L
			ALT	27	U/L	6 - 34
			AST	40	U/L	9 - 34 H
			BILI	0.6	mg/dl	0.2 - 1.2
			BUN	10	mg/dl	4 - 24
			CA	8.8	mg/dl	8.4 - 10.3
			CHOL	230	mg/dl	156 - 300
			CK	70	U/L	21 - 169
			CL	106	mEq/L	94 - 112
			CREAT	0.9	mg/dl	0.7 - 1.4
			GGT	15	U/L	5 - 50
			GLUC	85	mg/dl	50 - 250
			K	4.5	mEq/L	3.4 - 5.4
			NA	140	mEq/L	135 - 145
			PHOS	3.8	mg/dl	2.2 - 5.1
			PROT	6.1	g/dl	6 - 8
			URATE	4.5	mg/dl	2.5 - 7.5
4	WEEK 2	1/16/2014 1:17 PM	ALB	3.9	g/dl	3.3 - 4.9

Ready CHEMISTRY HEMATOLOGY OTHER URINALYSIS 100%

49

Here is the SAS output, created in real-time, and delivered to the client. Refer to the accompanying paper for more information about this topic.

## SAS/IntrNet Application Dispatcher

**SAS 9.3:** <http://tinyurl.com/3qpxzlg>

**SAS 9.4:** <http://tinyurl.com/n8uqq92>

## SAS Stored Processes

**SAS 9.3:** <http://tinyurl.com/3wyw6fk>

**SAS 9.4:** <http://tinyurl.com/q99jov6>

## Conclusion

- Use ExcelXP tagset to create XML file
- Resulting XML file can be viewed with Excel
- Make use of tagset options
- Apply ODS style overrides
- Use Excel formats instead of SAS formats



Using ODS to generate specially formatted XML output is an effective method of incorporating SAS output into Excel documents.

The SAS®9 ExcelXP ODS tagset provides an easy way to export your SAS data to Excel workbooks that contain multiple worksheets. By using ODS styles, style overrides, and a tagset that complies with the Microsoft XML Spreadsheet Specification, you can customize the output to achieve your design goals.

SAS Institute continues to work toward better Microsoft Office integration, and future releases of SAS software will provide even more robust means of using SAS content with Microsoft Office applications.

## Resources

- Paper & Download Package

[support.sas.com/rnd/papers/index.html#excel2014](http://support.sas.com/rnd/papers/index.html#excel2014)

- Vince's ExcelXP Resources

[www.sas.com/reg/gen/corp/867226?page=Resources](http://www.sas.com/reg/gen/corp/867226?page=Resources)



51



## Contact Information

Please send questions, comments and feedback to:

Vince DelGobbo  
sasvcd@SAS.com

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 ([support.sas.com/usergroups/namerica/lug-form.html](http://support.sas.com/usergroups/namerica/lug-form.html)) at least eight weeks in advance.

52

### About the author:

Vince DelGobbo is a Senior Software Developer in the Metadata and Execution Services group at SAS. This group's responsibilities include the SAS/IntrNet Application Dispatcher and SAS Stored Processes. He is involved in the development of new Web- and server-based technologies, as well as integrating SAS output with Microsoft Office. He was also involved in the early development of the ExcelXP ODS tagset. Vince has been a SAS Software user since 1982, and joined SAS in 1992.