

PharmaSUG 2019 - Paper AP-42

(Microsoft) Excel in SAS®

Charu Shankar, SAS® Institute Inc.

ABSTRACT

Excel and SAS are universally loved. Both have their strengths. Excel has been around a long time and many non-SAS users use the spreadsheet to enter & manage their transactions. SAS is great for analyzing data. Why not marry the strengths of both? get data from excel into SAS, complete the analysis in SAS and then send the results to excel. This would be a great help for colleagues who don't have SAS on their desktops. Come learn the many ways to get Excel to SAS. From excel and data from SAS to excel. This session will cover the following ways to move between Excel and SAS.

1. PROC IMPORT - to read Excel into SAS
2. SAS Access engines to read Excel into SAS
3. ODS Tagsets - take sas to Cool Excel pivot tables
4. PROC EXPORT - export SAS to Excel

DATA USED IN THIS PRESENTATION

Data can be categorized into 2 buckets: Structured and Unstructured. The technique that is used will mainly depend on the type of data that we need to read in.

TABLE NAME	DETAILS
Sashelp.cars	The Sashelp.cars data set provides 2004 car data. The data set contains 428 observations.
Sales.xlsx	Demographic information about sales employees in Australia and the United States
Sashelp.orsales	The Sashelp.orsales data set provides Orion star sports and outdoors sales 1999–2002. The data set contains 912 observations.
Class.xlsx	The Sashelp.Class data set provides information about a small fictitious class of students. Variables include Sex, Age, Height, and Weight. The data set contains 19 observations

Table 1. Details about the data sets used in this presentation

1. IMPORTING DATA INTO SAS

Importing unstructured data needs some special instructions and can be accomplished by using PROC IMPORT.

PROC IMPORT Syntax

```
PROC IMPORT DATAFILE="path/file-name.csv" DBMS=CSV
              OUT=output-table;
RUN;
```

Common DBMS identifiers that are included with Base SAS:

- CSV – comma-separated values.
- JMP – JMP files, JMP 7 or later.
- TAB – tab-delimited values.
- DLM – delimited files, default delimiter is a space. To use a different delimiter, use the DELIMITER= statement.

Additional DBMS identifiers included with SAS/ACCESS Interface to PC Files:

- XLSX – Microsoft Excel 2007, 2010 and later
- ACCESS – Microsoft Access 2000 and later

Other DBMS identifiers can be viewed [here](#) in the SAS Help Center.

The following code above will import the class_test sheet from the class excel workbook and create a SAS dataset called class_test_import in the WORK library.

Code to create Excel workbook from SAS data set

```
proc import datafile="s:/workshop/data/class.xlsx"
            dbms=xlsx
            out=work.class_test_import replace;
            sheet=class_test;
run;
```

2. SAS ACCESS ENGINES TO READ EXCEL INTO SAS

Structured data can be easily read with SAS access engines. This technique is so simple that it takes users' breath away whenever I demonstrate it in class. You may already have a long checklist of the things SAS can do. But, have you considered what I consider to be a great SAS strength? It's the ability to read virtually any data: be it excel, Teradata, oracle, db2, and the list goes on.

How does SAS read all these foreign files with minimum sweat? Engines allow you to point to these foreign files where your data is residing without paying any expensive import duty.

Magically, on the SAS side, these files will look like Sas data sets, so you can go ahead and analyze, manage, and report on these tables to your heart's content using SAS procedures. By the way, any SAS procedure is fair game for these files read via engines into SAS.

Here is the syntax to point to Excel using SAS access engines. Bear in mind that the engine & syntax may vary depending on the bitness of your operating environment.

Both SAS and Microsoft Office offer 32-bit and 64-bit versions.

[Bitness](#) refers to the 32/64 bit and version of software. see the following usage notes

- [Installing SAS® 9.3 PC Files Server and using it to convert 32-bit Microsoft Office files to SAS® 64-bit files](#)
- [Summary of steps to install and use the SAS® 9.4 PC Files Server](#)


SAS/ACCESS LIBNAME Statement


SAS/ACCESS has multiple LIBNAME engines that access Microsoft Excel workbooks.

```
libname orionx excel "&path\sales.xlsx";
```

```
libname orionx pcfiles path="&path\sales.xlsx";
```

```
libname orionx xlsx "&path\sales.xlsx";
```


 The engine that you use depends on the operating environment of SAS and the bitness of SAS and Excel.



The following table summarizes nicely the access engines in operating environments

SAS/ACCESS Engines

	EXCEL Engine	PCFILES Engine	XLSX Engine
SAS Operating Environment	Windows	Windows, Linux, and UNIX	Windows, Linux, and UNIX
Bitness	Same	Any combination	Any combination
Additional Software	Yes	Yes	No
Supported Excel File Types	.xls .xlsx	.xls .xlsx	.xlsx
Options	Yes	Yes	Limited



Let's examine the Sales.xlsx workbook. It has 2 sheets in it: Australia and United States.

Examining sales.xlsx

7

sas

Exploring the Library

Regardless of the LIBNAME engine that is used, you can use the CONTENTS procedure to explore the library.

```
libname orionx pcfiles path="&path\sales.xlsx" ;
proc contents data=orionx._all_ ;
run;
libname orionx clear;
```

- When SAS has a libref that is assigned to an Excel workbook, the workbook cannot be opened in Excel.
- To disassociate the libref, use a LIBNAME statement with the CLEAR option.

p107d01a
p107d01b sas

Exploring the Library

Partial PROC CONTENTS Output

The CONTENTS Procedure

#	Name	Member Type	DBMS Member Type
1	Australia\$	DATA	TABLE
2	UnitedStates\$	DATA	TABLE

EXCEL and PCFILES engines

Note: For the EXCEL and PCFILES engines, worksheet names end with a dollar sign.

p107d01a sas

Exploring the Library

Additional PROC CONTENTS Output

#	Variable	Type	Len	Format	Informat	Label
8	Birth_Date	Num	8	DATE9.	DATE9.	Birth Date
7	Country	Char	2	\$2.	\$2.	Country
1	Employee_ID	Num	8			Employee ID
2	First_Name	Char	10	\$10.	\$10.	First Name
4	Gender	Char	1	\$1.	\$1.	Gender
9	Hire_Date	Num	8	DATE9.	DATE9.	Hire Date
6	Job_Title	Char	14	\$14.	\$14.	Job Title
3	Last_Name	Char	12	\$12.	\$12.	Last Name
5	Salary	Num	8			Salary

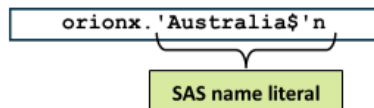
Note: Each engine supplies formats and labels for the Excel data.

22

sas

SAS Name Literal

Because the EXCEL and PCFILES engines refer to worksheets with dollar signs at the end of the names, use a SAS name literal to permit the special character in the worksheet name.



The XLSX engine does not refer to worksheets with a dollar sign at the end of the name, so a SAS name literal is not needed.

orionx.Australia

23

sas

Subsetting a Worksheet

Regardless of the LIBNAME engine used, the PRINT procedure can be used to display a subset of the worksheet.

```
proc print data=orionx.'Australia$n' noobs;
  where Job_Title contains 'IV';
  var Employee_ID Last_Name
      Job_Title Salary;
run;
```

EXCEL and
PCFILES engines

```
proc print data=orionx.Australia noobs;
  where Job_Title contains 'IV';
  var Employee_ID Last_Name
      Job_Title Salary;
run;
```

XLSX
engine

24

p107d02a
p107d02b

sas

There you go. Pointing to Excel with the SAS access engine makes SAS treat it as if it's a SAS dataset. You can do so much more with this technique.

3. ODS TAGSETS - TAKING SAS TO COOL EXCEL PIVOT TABLES

This is a technique that always draws Oohs & Aahs every time I teach. It's the ability of SAS code to draw up and shape a pivot table right under your eyes.

The concept and code for this is so simple and yet, many users find it intimidating. Let's break this down into simple steps.

I like to call this the sandwich technique. Just like a sandwich has a piece of bread and a filling, the bread layer is your destination: pdf, excel etc. and the inner filling is your procedure. Here is my garden party from last summer. Do you like the eggplant sandwich?



A pivot table

A *pivot table* is an interactive, cross-tabulated Excel report that summarizes and analyzes data from various sources. The benefits are: Drill-down capabilities, User-friendly, hands-on reports, Pivot tables can be performed on multiple subsets of data.

Updating the Template Item Store

By default, templates are stored in the SASUSER.TEMPLAT item store unless the ODS PATH is modified. Errors might occur when writing to the SASUSER location if the access mode for the template store has been changed to (read). To eliminate error messages, use the ODS PATH statement to write to an alternate location.

The example below adds a temporary item store in the Work directory where templates will be written. In this example, the work.templat item store is added before the default path.

```
ods path(Prepend) work.templat(update) ;
```

```
ods tagsets.tableeditor  
file="%sysfunc(getoption(work)) \pharmasug.html"
```

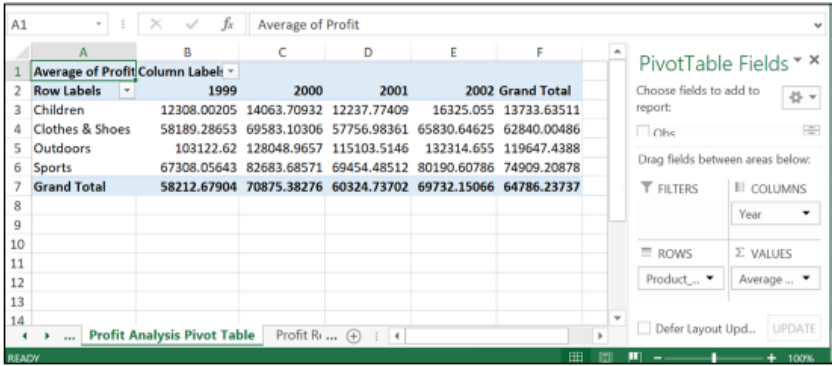
%sysfunc(getoption(work)) points to the location of the **Work** library.

Code to create Excel Pivot tables

```
ods tagsets.tableeditor
file="%sysfunc(getoption(work))\pharmasug.html"
options(
pivotrow="product_line"
pivotcol="year"
pivotdata="profit"
pivot_sheet_name="Profit Analysis Pivot Table"
sheet_name="Profit Report"
pivotdata_stats="average"
pivotcharts="yes"
chart_type="conecol"
doc='help' );
proc print data=sashelp.orsales;
run;
ods tagsets.tableeditor close;
```

Voila! Your pivot table is ready!

Pivot Table Ready



Average of Profit	Column Labels	1999	2000	2001	2002	Grand Total
Children		12308.00205	14063.70932	12237.77409	16325.055	13733.63511
Clothes & Shoes		58189.28653	69583.10306	57756.98361	65830.64625	62840.00486
Outdoors		103122.62	128048.9657	115103.5146	132314.655	119647.4388
Sports		67308.05643	82683.68571	69454.48512	80190.60786	74909.20878
Grand Total		58212.67904	70875.38276	60324.73702	69732.15066	64786.23737

SAS

Tip sheet of Pivot table options

Tip Sheet of Pivot Table Options

Add pivot tables to an Excel worksheet using the TableEditor tagset.

Use options such as PIVOTROW=, PIVOTCOL=, PIVOTPAGE, and PIVOTDATA= to add one or more columns to a row, a column, a page, and data areas of the layout in Excel.

Add multiple variables with one of these options by separating each variable with a comma.

Use additional options to modify statistics from the default of SUM.

A button is added by default to the Web page.

When using Microsoft Internet Explorer, the button can be selected to create the pivot table on Microsoft Windows.

The TableEditor tagset can be downloaded from [this location](#).

4. PROC EXPORT – EXPORT SAS TO EXCEL

Finally, let's take a peak at another super easy way to send SAS to excel. This is similar to the first technique in this paper of using a SAS procedure to export data.

PROC EXPORT Syntax

tells SAS how to format the output

```
PROC EXPORT DATA=input table OUTFILE="output-file"
  <DBMS=identifier> <REPLACE>;
RUN;
```

The following code above will export the Cars dataset in the SASHELP library to an excel workbook in the desired location.

Code to create Excel workbook from SAS data set

```
proc export data=sashelp.cars
  outfile="c:/workshop/pharmasug/output/cars.xlsx"
  dbms=xlsx replace;
run;
```


5. REFERENCES

"Importing a .xlsx file into SAS". Support.sas.com website.

https://sas.okta.com/app/sasinstituteinc_communitiessascom_1/exk3b9v7kyEu1p8Lw2p7/sso/saml

"PROC IMPORT". SAS® 9.4 and SAS® Viya® 3.4 Programming Documentation-Base SAS Procedures Guide

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n1qn5sclnu2l9dn1w6l1fw8wqhts.htm&locale=en

"PROC EXPORT". SAS® 9.4 and SAS® Viya® 3.4 Programming Documentation-Base SAS Procedures Guide

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=proc&docsetTarget=n045uxf7ll2p5on1ly4at3vpd47e.htm&locale=en

"LIBNAME Statement". Support.sas.com website

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=lestmtsglobal&docsetTarget=n1nk65k2vsfmxfn1wu17fntzszbp.htm&locale=en

"SAS/ACCESS libname engines".

https://go.documentation.sas.com/?cdcId=pgmsascdc&cdcVersion=9.4_3.4&docsetId=apcpcref&docsetTarget=p0xxl8274gj8j6n189y1ij80kv8z.htm&locale=en

"Using SAS to add Pivot Tables to your Excel Workbook". Blogs.sas.com
Parker, Chevell. March 2015

<https://blogs.sas.com/content/sgf/2015/03/27/using-sas-to-add-pivottables-to-your-excel-workbook/>

"How SAS engines can read a range of data in Excel". Blogs.sas.com
Shankar, Charu. November 2010

<https://blogs.sas.com/content/sastraining/2010/11/29/how-sas-engines-can-read-a-range-of-data-in-excel/>

6. ACKNOWLEDGMENTS

The author is grateful to the many SAS users that have entered her life. With every new question asked, she has been able to research and study new ways to express the fabulous ways in which SAS and Excel talk to each other. Sometimes a user appeared in the form of a teacher to show her the many ways in which to express the Sas language. The users are too many to thank individually so this is a thank you to every single user who has touched her life. This paper was originally an invited presentation at SAS global forum. Charu is grateful to the Pharnasug committee for inviting her to present this paper. She would also like to express her gratitude to her manager, Stephen Keelan without whose support and permission, this paper would not be possible.

7. CONTACT INFORMATION

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

Charu Shankar
SAS Institute Canada, Inc.
Charu.shankar@sas.com
<https://blogs.sas.com/content/author/charushankar/>

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.